

Technische Universität München  
Lehrstuhl für Mensch-Maschine-Kommunikation

# Methoden und Anwendungen zur intuitiven Mensch-Roboter-Interaktion

Jürgen Blume

Vollständiger Abdruck der von der Fakultät für Elektrotechnik und Informationstechnik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs (Dr.-Ing.)

genehmigten Dissertation.

Vorsitzender: Univ.-Prof. Dr.-Ing. Georg Sigl

Prüfer der Dissertation: 1. Univ.-Prof. Dr.-Ing. habil. Gerhard Rigoll

2. Univ.-Prof. Dr.-Ing. habil. Dr. h.c. Alexander W. Koch

Die Dissertation wurde am 29.04.2014 bei der Technischen Universität München eingereicht und durch die Fakultät für Elektrotechnik und Informationstechnik am 01.12.2014 angenommen.



## **Zusammenfassung**

Das Ziel der vorliegenden Arbeit ist, durch die Verwendung aktueller Interaktionstechnologie und Methoden eine intuitive Interaktion zwischen Mensch und Roboter mit dem Fokus auf Benutzerschnittstellen und Bedienbarkeit zu erreichen. Dabei wurden unterschiedliche Aspekte der Interaktion zwischen Mensch und Roboter sowohl im sozialen als auch im industriellen Umfeld betrachtet. Dies führte zu der Entwicklung von Interaktionskonzepten, Prototypen und der Evaluierung im Rahmen von Nutzerstudien.

Hierfür wurde zunächst eine multimodale Architektur zur Interaktion konzipiert und in einem interdisziplinären Forschungsverbund um multimodale Interaktionsmöglichkeiten erweitert. Dabei kam für die Ablaufsteuerung in der Architektur eine wissensbasierte Kontrollinstanz zur Anwendung. Für eine natürliche Interaktion mit einer mobilen Roboterplattform in Dialogsituationen wurde eine Blicksteuerung entwickelt, die an menschliches Verhalten angelehnt ist. Eine intuitive Interaktion wurde auch im Rahmen der Programmierung von Robotern betrachtet und evaluiert. Die Arbeit schließt mit einer Übersicht über die realisierten Anwendungsszenarien für eine intuitive Interaktion im sozialen und industriellen Umfeld.



## **Abstract**

The aim of this thesis is to provide an intuitive interaction between humans and robots by using recent interaction technology and methods focusing on user interfaces and usability. Different aspects of the interaction between human and robot in social and industrial environments were studied and concepts, first prototypes and evaluations in user studies were developed.

For this purpose an architecture supporting multimodal interaction was designed and various interaction channels were created in collaboration within an interdisciplinary research cluster. The orchestration of the components in the architecture was realized using a knowledge-based system controller. For a natural interaction with a mobile robotic platform in dialogue situations, a human-inspired gaze control was developed. An intuitive interaction was also considered and evaluated for programming robots. The thesis concludes with an overview of the application scenarios for intuitive interaction in social and industrial environments.



## Danksagung

Diese Arbeit wäre ohne die Unterstützung vieler Personen nicht möglich gewesen. Auch wenn es unmöglich ist, jedem Einzelnen an dieser Stelle zu danken, so möchte ich doch zumindest einigen stellvertretend meine Dankbarkeit ausdrücken. Zunächst möchte ich mich bei Herrn Professor Gerhard Rigoll für die Möglichkeit zur und Unterstützung bei der Promotion bedanken. Ein weiterer Dank gilt Professor Frank Wallhoff für die Koordination der Projekte, die den Rahmen für die Inhalte dieser Arbeit maßgeblich mitgeprägt haben. Weiterer Dank für die administrative und technische Unterstützung gilt Peter Brand, Heiner Hundhammer, Wibke Borngesser, Ulrike Scholz, Melitta Schubert, Martina Römpf, Carola Zwicker, Eric Bourguignon und Dr. Uwe Haass.

Danke für die fruchtbare Zusammenarbeit und die Erstellung diverser Demonstratoren im interdisziplinären Forschungsverbund an alle Beteiligten, jedoch insbesondere an Florian Rohrmüller, Daniel Althoff, Stefan Sosnowski, Christoph Mayer, Claus Lenz. Bei Stefan Kohlbecher möchte ich mich an dieser Stelle für die Unterstützung bezüglich der Blickerfassung und für die unvergesslichen *Augenblicke* mit den Robotern bedanken.

Danke an alle Studenten, Lehrstuhl- und Projektkollegen, die durch Diskussion, Meinungen und andere Beiträge zum Gelingen dieser Arbeit beigetragen haben. Ganz besonderen Dank verdienen Alexander Bannat, Tobias Rehrl und Wolfgang Rösel für die schöne Zeit am und außerhalb des Lehrstuhls sowie für die ununterbrochene Motivation und Unterstützung bei unzähligen Herausforderungen. Bezüglich Korrekturen gebührt der Dank unter anderem Alexander Bannat, Tobias Rehrl, Jürgen Geiger, Christoph Mayer und Claus Lenz.

An dieser Stelle möchte ich auch meinen Eltern Ingrid und Dieter sowie meinen Brüdern für ihre Unterstützung und meinen ersten PC danken. Neben meiner Gastfamilie möchte ich auch meiner Blumenfamilie für jegliche Unterstützung danken. Auch ein Dank an Familie Bauer für vieles, aber ganz besonders für das Kochen.

Mein herzlichster Dank gebührt meiner Frau Barbara für das Ertragen meiner Stimmungsschwankungen während der Erstellung dieser Arbeit sowie für den Korrekturmarathon.



# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Motivation . . . . .	1
1.2	Beitrag der Arbeit . . . . .	4
1.3	Aufbau der Arbeit . . . . .	5
<b>2</b>	<b>Mensch und Roboter</b>	<b>7</b>
2.1	Entwicklung . . . . .	7
2.2	Gemeinsamer Arbeitsbereich . . . . .	9
2.3	Vision: Roboter als Assistent in einer industriellen Umgebung . . . . .	13
<b>3</b>	<b>Theoretische und methodische Grundlagen</b>	<b>19</b>
3.1	Theoretische Grundlagen . . . . .	19
3.2	Benutzerschnittstellen . . . . .	27
3.3	Evaluierung . . . . .	29
3.4	Verwendete Software . . . . .	32
<b>4</b>	<b>Architektur zur multimodalen Mensch-Roboter-Interaktion</b>	<b>35</b>
4.1	Motivation . . . . .	35
4.2	Verwandte Arbeiten . . . . .	36
4.3	Multimodale Interaktion . . . . .	38
4.4	Architektur zur multimodalen Interaktion . . . . .	49
4.5	Zusammenfassung . . . . .	53

<b>5</b>	<b>Blicksteuerung in Dialogsituationen</b>	<b>55</b>
5.1	Motivation . . . . .	55
5.2	Verwandte Arbeiten . . . . .	57
5.3	Modell für Blickverhalten . . . . .	60
5.4	Relevante Beobachtungen . . . . .	62
5.5	Implementierung und Ergebnisse . . . . .	66
5.6	Zusammenfassung . . . . .	74
<b>6</b>	<b>Lernmethoden in der Mensch-Roboter-Kooperation</b>	<b>75</b>
6.1	Überblick über Lernverfahren für Roboter . . . . .	76
6.2	Instruktionsbasiertes Lernen für hybride Montageprozesse . . . . .	83
6.3	Lernen von Verpackungsschritten aus Punktwolken . . . . .	87
6.4	Intuitive Programmierung einer Verpackungstation . . . . .	103
6.5	Zusammenfassung . . . . .	110
<b>7</b>	<b>Anwendungen</b>	<b>111</b>
7.1	Roboter- und Forschungsplattformen . . . . .	111
7.2	Soziales Umfeld . . . . .	116
7.3	Industrielles Umfeld . . . . .	120
<b>8</b>	<b>Zusammenfassung und Ausblick</b>	<b>129</b>
8.1	Zusammenfassung . . . . .	129
8.2	Ausblick . . . . .	130
	<b>Glossar</b>	<b>131</b>
	<b>Abkürzungsverzeichnis</b>	<b>135</b>
	<b>Abbildungsverzeichnis</b>	<b>139</b>
	<b>Tabellenverzeichnis</b>	<b>143</b>
	<b>Literaturverzeichnis</b>	<b>145</b>

# Kapitel 1

## Einleitung

### 1.1 Motivation

Roboter sind heutzutage nicht mehr aus der Automatisierungsindustrie wegzudenken. Schon seit 1960 werden Roboter in der Industrie eingesetzt, um Handhabungen zu erledigen, die sich immer auf die gleiche Art und Weise ausführen lassen. Der Roboter kann Aufgaben mit besonders großer Präzision erledigen und schwere Lasten (beispielsweise ganze Fahrzeuge) heben. Zusätzlich ist für die Industrie attraktiv, dass der Roboter – abgesehen von Wartung und Reparatur – 24 Stunden am Tag und sieben Tage die Woche seinen vorgesehenen Dienst verrichten kann.

Laut einem Bericht [1] der International Federation of Robotics (IFR) schafft der Einsatz von Robotern in der Produktion insgesamt sogar mehr Arbeitsplätze, obwohl einfache sowie unangenehme und repetitive Tätigkeiten für den Menschen wegfallen. Laut dem Bericht werden weltweit bis zum Jahr 2016 durch den Einsatz von Robotern voraussichtlich mehr als eine Million Arbeitsplätze geschaffen werden. Der Bericht korreliert für diese Prognose den Erwerb neuer Roboter im Land mit der Entwicklung der Beschäftigtenzahlen.

Der Einsatz von Robotern in der Industrie kann auch den Trend zum Verlagern der Produktion ins Ausland zumindest dämpfen. Dadurch können Teile der Produktion und damit verbundene Arbeitsplätze im Inland trotz steigendem Kostendruck gehalten werden. Dies geht beispielsweise aus einer Studie [2] der dänischen Zeitung *Ugebladet A4* hervor. Von den 213 befragten dänischen Betrieben aus der Fertigungsbranche hatten bereits 58 % Roboter im Einsatz und wiederum 84 % der Unternehmen mit Robotern gaben an, ihre Wettbewerbsfähigkeit sei durch den Einsatz von Robotern stark

## 1. EINLEITUNG

---



**Abbildung 1.1:** Produktvielfalt am Beispiel eines Loewe Fernsehers Individual [3]

bis sehr stark gestiegen. Außerdem gaben 31 % an, durch Robotereinsatz sei die Notwendigkeit für eine Auslagerung der Produktion zurückgegangen, während nur 2 % der Unternehmen mit Robotern eine stärkere Notwendigkeit für die Auslagerung sahen.

Die Entwicklung in der Industrie, verstärkt auf Kundenwünsche einzugehen und Produkte anzubieten, die der Kunde individualisieren kann, sowie immer kürzere Produktzyklen und die Effekte der Globalisierung stellen sowohl neue Herausforderungen als auch Chancen für die zukünftigen Automatisierungsprozesse dar. In Abbildung 1.1 sind beispielsweise drei unterschiedliche Farben und Aufstellungsvarianten eines Fernsehers zu sehen. Dieses Produkt gibt es in über zwölf Farben, aus denen der Kunde wählen kann. Zusammen mit weiteren Konfigurationsoptionen (Technikausstattung, Rahmentyp und -farbe, Bildschirmdiagonale, Aufstellungsvariante etc.) [3] kann der Käufer insgesamt 2160 verschiedene Varianten dieses Fernsehermodells erstellen. Bei der Fertigung kann die Handhabung der schweren Geräte (bis zu 30 kg) durch Service-roboter den Werker bei seiner Arbeit entlasten.

Anwendungen für Servicerobotik und Untersuchungen bezüglich ihrer Wirtschaftlichkeit sowie deren weitere Bedeutung für die Robotik-Entwicklung sind Gegenstand der Studie in [4]. Dafür analysieren das Fraunhofer Institut Produktionstechnik und Automatisierung (IPA) und das Fraunhofer Institut für System- und Innovationsforschung (ISI) das Potenzial solcher neuen Einsatzfelder, beauftragt vom Bundesministerium für Bildung und Forschung (BMBF). Einer der betrachteten Anwendungsfälle ist ein sogenannter Produktionsassistent, der den Werker bei seiner Arbeit unterstützen soll. Hierbei wird bei den identifizierten Forschungsbedarfen neben der sicheren Interaktion, Objekterkennung und angepasster Roboterhardware auch die Art und Weise der Kommunikation für eine interaktive Steuerung des Roboters herausgestellt. Für



**Abbildung 1.2: Tablet-Steuerung von Robotern** - (a) ReisPad [5] für Industrieroboter der Firma Reis Robotics (b) Tablet zur Steuerung eines Roboters von Universal Robots [6]

die Interaktion soll dabei insbesondere eine bidirektionale Kommunikation über den Sprachkanal realisiert werden, wobei eine potenzielle Fehlbedienung verhindert werden soll. Außerdem soll auch die Erkennung von Gesten und Bewegungen des Werkers unterstützt werden, um Handlungsanweisungen geben zu können beziehungsweise Arbeitsbereiche korrekt zu interpretieren.

Ein weiterer Vorteil einer intuitiven Interaktion zur einfacheren Programmierung von Robotern ist, dass Fachkräfte, die durch den Einsatz von Robotern bei unangenehmen Tätigkeiten unterstützt werden, die Möglichkeit haben, diese neuartigen Roboter selbst zu programmieren oder zu adaptieren. Schließlich wissen die Fachkräfte in der Produktion am besten, wo Probleme beziehungsweise Optimierungsmöglichkeiten im Prozessablauf bestehen. Für die Industrie lassen sich durch die einfachere Programmierung auch Zeit und Kosten einsparen, da somit keine gut bezahlten Experten für jeden Programmiervorgang benötigt werden. Natürlich fallen jedoch weiterhin die Wartungs- und Instandhaltungskosten für das Produktionssystem an.

Zudem haben auch neue Technologien und Geräte, wie beispielsweise Tablets, Smartphones, Tiefenbildkameras sowie Sprach- und Gestenerkennung das Feld der natürlichen und intuitiven Bedienbarkeit verändert und bereichert. Diese neuen Bedienmöglichkeiten bieten auch neue Perspektiven und Anwendungsmöglichkeiten im Bereich der Mensch-Roboter-Interaktion. Ein Beispiel hierfür ist das ReisPad, das die

Firma Reis Robotics, inspiriert vom großen Erfolg der Tablets, als berührungssensitives Eingabegerät [5] für ihre Roboter entwickelt hat und das in Abbildung 1.2 dargestellt ist. Auch der Roboterhersteller Universal Robotics, der sich auf nachgiebige Roboter für den Einsatz in der Nähe des Menschen spezialisiert hat, benutzt Tablets als Bedien- und Programmiergerät [6], um eine einfache Inbetriebnahme der Roboter zu gewähren.

### **1.2 Beitrag der Arbeit**

Diese Arbeit beschäftigt sich mit Aspekten der Interaktion zwischen Mensch und Roboter. Dabei werden unterschiedliche Themengebiete betrachtet. Es wird eine Architektur für die multimodale Interaktion vorgestellt, die sowohl in einem industriellen als auch in einem sozialen Umfeld prototypisch zur Anwendung gebracht wurde. Dabei wurden Aspekte der Realzeitverarbeitung sowie einer auf Expertenwissen basierten Kontrollinstanz berücksichtigt. Ein Modell für eine Blicksteuerung für Roboter in Dialogsituationen wurde entworfen. Dabei wurden visuelle sowie auditive Reize berücksichtigt. Das Modell wurde prototypisch auf einer Roboterplattform implementiert und evaluiert.

Intuitive Interaktion zwischen Mensch und Roboter wurde auch bezüglich einer einfacheren Programmierung von Industrierobotern untersucht. Dafür wurden unterschiedliche Ansätze zur Programmierung betrachtet. Neben einer Möglichkeit das System über Instruktionen zu programmieren, wurde auch das Programmieren durch Vormachen mittels der Analyse von Punktwolkendaten studiert sowie ein kombinierter Ansatz verfolgt und um eine visuelle Repräsentierung der Arbeitsschritte ergänzt. Für die prototypische Implementierung wurden Smartphones und Tablets sowie Tiefenbildkameras und aktuelle Spracherkennungstechnologien verwendet. Insbesondere war dabei nicht nur der Roboter programmierbar, sondern es konnten auch Aktionen des mit dem Roboter zusammenarbeitenden Menschen hinterlegt werden. Durch die Miteinbeziehung der Aktionen des Menschen in den Arbeitsablauf kann die Interaktion zwischen Mensch und Roboter besser synchronisiert werden. Die verwendeten Interaktionskanäle und Bedienmedien des Prototyps wurden im Rahmen einer Nutzerstudie evaluiert und überzeugten bezüglich der pragmatischen sowie der hedonischen Qualität.

## 1.3 Aufbau der Arbeit

Im Folgenden wird der weitere Aufbau der vorliegenden Arbeit kurz erläutert. Das Kapitel 2 präsentiert einen Überblick über die Entwicklung der Zusammenarbeit zwischen Mensch und Roboter. Nach einer kurzen Übersicht über die Entwicklung wird hier der gemeinsame Arbeitsbereich für eine Mensch-Roboter-Kooperation erläutert. Das Kapitel schließt mit der Präsentation einer Vision des Roboters als Assistent in der Montage. Das Kapitel 3 präsentiert einige Grundlagen bezüglich der verwendeten mathematischen Verfahren, einen kurzen Überblick über die angewendeten Evaluierungsmethoden und endet mit einer Übersicht der verwendeten Softwarepakete, die zur Entwicklung der Prototypen eingesetzt wurden. Im Kapitel 4 werden die unterschiedlichen Kommunikationskanäle der multimodalen Interaktion zwischen Mensch und Roboter eingeführt und eine dazu entwickelte Architektur präsentiert. Im Anschluss folgt das Kapitel 5, in dem eine Blicksteuerung für Roboter in Dialogsituationen vorgestellt wird. Das Kapitel 6 handelt von Lernmethoden für die Mensch-Roboter-Kooperation. Dabei werden zunächst unterschiedliche Lernmethoden aufgezeigt und dann die entwickelten Konzepte präsentiert. Darauf folgen die realisierten Anwendungen in Kapitel 7. Es werden die verwendeten Forschungsplattformen und Roboterplattformen beschrieben sowie die auf den vorherigen Kapiteln basierenden Anwendungen im sozialen und industriellen Umfeld beschrieben. Die Arbeit wird in Kapitel 8 mit einer Zusammenfassung rekapituliert und schließt mit einem Ausblick auf weitere Entwicklungen in der Mensch-Roboter-Interaktion.



# Kapitel 2

## Mensch und Roboter

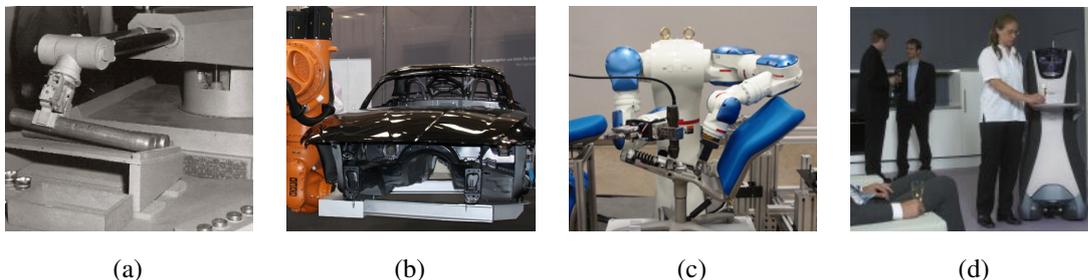
Dieses Kapitel beschäftigt sich zu Beginn mit der Entwicklung von Robotern im industriellen Einsatz sowie dem zunehmenden Trend hin zur Servicerobotik im sozialen Umfeld. Nach einem kurzen Überblick über die bestehenden Herausforderungen in Bezug auf die Zusammenarbeit zwischen Mensch und Roboter, folgt die Vision des Roboters als Assistent in industriellen Umgebungen. Dabei werden auch weitere Projekte, die sich mit diesem Forschungsthema beschäftigen, präsentiert.

### 2.1 Entwicklung

Laut [7] lassen sich die ersten Ideen zu intelligenten Mechanismen, die wir heute Roboter nennen würden, bereits circa 3500 vor Christus in griechischen Mythologien finden. Es folgten noch viele weitere Ideen, Konzepte und Prototypen, die in [7] nachzulesen sind. Der Begriff „Robot“ wurde das erste Mal in dem Drama R.U.R. (Rossum's Universal Robots) [8] des tschechischen Schriftstellers Karel Čapek aus dem Jahr 1921 verwendet und erlangte durch die Kurzgeschichten von Isaac Asimov, wie beispielsweise Runaround [9] mit den drei Robotergesetzen, weitere Bekanntheit und Verbreitung. Einer der ersten industriellen Roboter der Firma Unimation wurde im Jahr 1959 entwickelt [10] und ist in Abbildung 2.1 (a) beim Einsatz im Jahre 1961 in einem Druckgusswerk von General Motors zu sehen. In der deutschen Automobilindustrie kamen die ersten Roboter im Jahr 1971 in einer Schweißtransferstraße [11] bei der Daimler Benz AG zur Anwendung. Die Roboter wurden von der Firma KUKA Roboter GmbH geliefert. Inzwischen können die Roboter dieser Firma ganze Fahrzeugkarossen handhaben, wie das Messe-Exponat eines KUKA KR 1000 TITAN [12] in Abbildung 2.1 (b) demonstriert.

## 2. MENSCH UND ROBOTER

---



**Abbildung 2.1: Roboterentwicklung** - (a) Roboter Unimate der Firma Unimation [10] (b) Roboter KUKA KR 1000 TITAN [12] mit Fahrzeugkarosserie (c) Zweiarmlroboter SDA10 von Motoman aus [14] (d) Fraunhofer Care-O-Bot 3 [15]

Die VDI-Richtlinie VDI 2860 [13] definiert Industrieroboter formal als

„[...] universell einsetzbare Bewegungsautomaten mit mehreren Achsen, deren Bewegungen hinsichtlich Bewegungsfolge und Wegen bzw. Winkeln frei (d. h. ohne mechanischen Eingriff) programmierbar und gegebenenfalls sensorgeführt sind. Sie sind mit Greifern, Werkzeugen oder anderen Fertigungsmitteln ausrüstbar und können Handhabungs- und/oder Fertigungsaufgaben ausführen.“

Roboter in der Industrie verfügen teilweise bereits über zwei Manipulatoren zur besseren Handhabung, wie der SDA10 von Motoman, der in Abbildung 2.1 (c) aus [14] bei der Montage eines Stuhls zu sehen ist. Laut dem Bericht [16] der IFR aus dem Jahr 2013 wurden im Verlauf des Jahres 2012 weltweit über 160 000 neue Industrieroboter verkauft. Der Großteil dieser Roboter wurde weiterhin an die Automobil- und die Elektronikindustrie verkauft. Allein für die Fertigung des Audi A3 Modells aus dem Frühjahr 2013 kommen in der dafür neu konstruierten Halle beispielsweise über 800 Roboter zum Einsatz [17]. Roboter kommen aber nicht mehr nur für Manipulationen in Fertigungsprozessen zum Einsatz, sondern können beispielsweise als mobile Plattformen Aufgaben in Reinräumen [18] übernehmen oder mittels Sensoren präzise optische Messungen [19] durchführen. Des Weiteren haben Roboter neben dem rein industriellen Einsatz im Laufe der Zeit weitere Anwendungsfelder erobert. Ein zunehmend wichtigeres und wachstumsstarkes Marktsegment für Roboterhersteller ist neben Spielzeug- [20, 21] und militärischen Robotern [22] beispielsweise die Servicerobotik. Inzwischen gibt es Serviceroboter, die Staub saugen [23], Fenster reinigen [24], den Rasen mähen [25], in der Landwirtschaft helfen, im Pflegebereich durch Tragen

der Patienten [26] oder durch den Transport von Getränken unterstützen, wie in Abbildung 2.1 (d), in der der Fraunhofer Care-O-Bot bei der Arbeit zu sehen ist. Eine weitere exzellente Roboterplattform ist der Roboter Justin [27] vom Deutschen Zentrum für Luft- und Raumfahrt.

Außerdem können Roboter für Such- und Rettungsmissionen [28] eingesetzt werden, wobei sie in für die Rettungsmannschaften extrem gefährliche Bereiche vordringen und weitere Informationen liefern können. Des Weiteren wird an Robotern geforscht, die die vorteilhaften Eigenschaften und Entwicklungen der Serviceroboter wieder zurück in die Fabriken bringen. In der Industrie können diese eingesetzt werden, um den Menschen bei schweren oder ermüdenden Aufgaben zu entlasten beziehungsweise zu unterstützen und um somit gemeinsam Aufgaben effizienter zu bewältigen. Eine nützliche Entwicklung der Servicerobotik ist beispielsweise die sichere Zusammenarbeit mit dem Menschen. Während die meisten Industrieroboter auch heutzutage meist noch aus Sicherheitsgründen isoliert vom Menschen die Aufgaben mit hoher Kraft und großer Geschwindigkeit verrichten, sind viele Serviceroboter bereits für den gemeinsamen Einsatz mit dem Menschen konzipiert.

Zum Verständnis der besonderen Herausforderungen einer Zusammenarbeit zwischen Mensch und Roboter in einer hybriden Fertigungsstation wird nachfolgend dieser gemeinsame Arbeitsbereich betrachtet.

## 2.2 Gemeinsamer Arbeitsbereich

Der gemeinsame Arbeitsbereich für Mensch und Roboter wird hier am Beispiel der industriellen Montage untersucht. Diese Montage kann man grundsätzlich in drei verschiedene Arten einteilen, nämlich in manuelle, hybride und vollautomatisierte Montage. Eine weitere Aufgliederung dieser drei Arten gibt der sogenannte Level of Automation (LoA) [29] an. Dabei wird neben vollkommen manuell und vollautomatisiert noch feiner unterteilt in insgesamt sieben Level. In Tabelle 2.1 sind die Level von eins „vollkommen manuell“ bis sieben „vollkommen automatisch“ dargestellt. Diese Level geben für Werkzeuge beziehungsweise Arbeitsstationen an, wie stark diese automatisiert sind. In diesem Zusammenhang ist auch der Modus für die Kontrolle des Systems definiert worden.

Ausgehend von dieser Tabelle ist die manuelle Montage in den Leveln eins bis vier anzusiedeln, während die hybride Fertigung den Leveln fünf bis sechs (flexible Arbeitsstation) entspricht. Der höchste Level sieben bezeichnet hingegen die vollautomatisierte Fertigung.

## 2. MENSCH UND ROBOTER

---

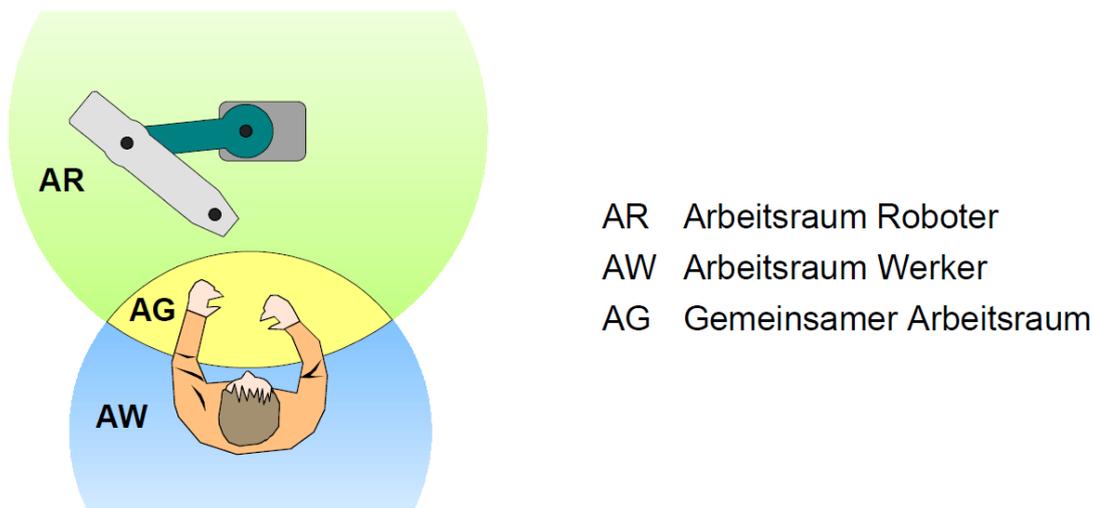
<b>Level</b>	<b>Werkzeuge</b>	<b>Kontrollmodus</b>
1	vollkommen manuell	vollkommen manuell
2	statisches Handwerkzeug	entscheidungsgebend
3	flexibles Handwerkzeug	lehrend
4	automatisches Handwerkzeug	fragend
5	statische Arbeitsstation	überwachend
6	flexible Arbeitsstation	eingreifend
7	vollkommen automatisch	vollkommen automatisch

**Tabelle 2.1:** Automatisierungslevel (engl: Level of Automation, LoA) [29] für Werkzeuge und Kontrollmodus

In dieser Arbeit wird der Fokus für das industrielle Umfeld auf die hybride Montage gelegt, da bei den anderen Montagearten eine natürliche Interaktion zwischen Mensch und Roboter so gut wie keine Rolle spielt. Des Weiteren können bei dieser Art der Montage die Stärken beider Partner genutzt werden. Der Mensch ist sehr flexibel einsetzbar und gut im Lösen von Aufgaben und Beheben von Fehlern. Der Roboter ist extrem präzise, kräftig und ermüdet nicht. Für die Interaktionspartner bei der hybriden Montage werden Arbeitsräume festgelegt. Im Gegensatz zur manuellen Montage gibt es jedoch nicht nur den Arbeitsraum des Werkers (AW) oder, wie bei der vollautomatisierten Montage den Arbeitsraum des Roboters (AR), sondern zusätzlich auch noch den gemeinsamen Arbeitsraum (AG). Dies ist sehr gut in Abbildung 2.2 von Thieme-rann [30] zu sehen.

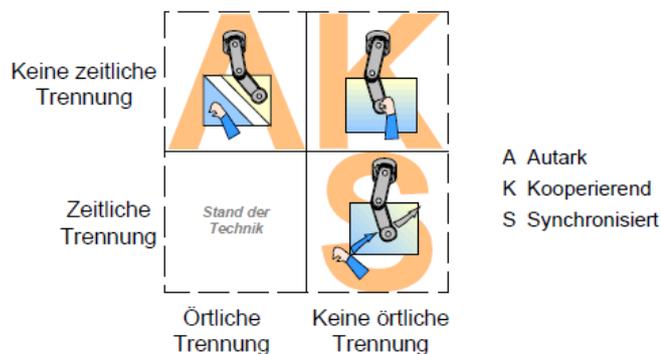
Eine frühe Aufstellung der Stärken von Mensch und Maschine hat Fitts bereits 1951 in der nach ihm benannten Liste [31] gegeben. Auch wenn nach über einem halben Jahrhundert der Entwicklung sicher nicht mehr alle Punkte dieser Liste gültig sind, so handelt es sich hierbei doch um eine Pionierarbeit auf diesem Gebiet. Des Weiteren werden die grundlegenden Stärken von Mensch und Maschine sehr gut erfasst, weshalb diese Liste auch heute noch häufig zur Aufgabenverteilung basierend auf den gelisteten Fähigkeiten herangezogen wird.

Wenn Mensch und Roboter eng zusammenarbeiten sollen, stellt sich die Frage nach der Sicherheit im Arbeitsbereich. Die Kombinationsmöglichkeiten der zeitlichen und örtlichen Trennung sind in Abbildung 2.3 als Matrix dargestellt. Hierbei kann der Arbeitsbereich beispielsweise örtlich getrennt werden, allerdings kann dann nicht mehr von einem echten gemeinsamen Arbeitsbereich gesprochen werden, weshalb dieser Ansatz für diese Arbeit nicht weiter von Bedeutung ist. Eine weitere Methode in der

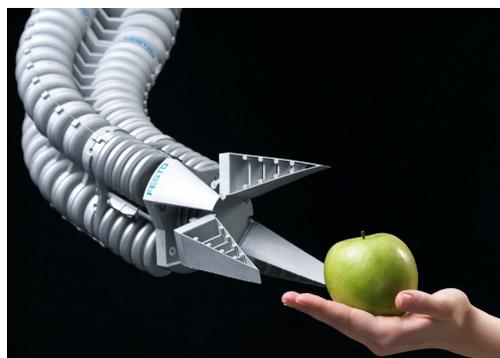


**Abbildung 2.2:** Gemeinsamer Arbeitsbereich von Werker und Roboter in der hybriden Montage aus [30]

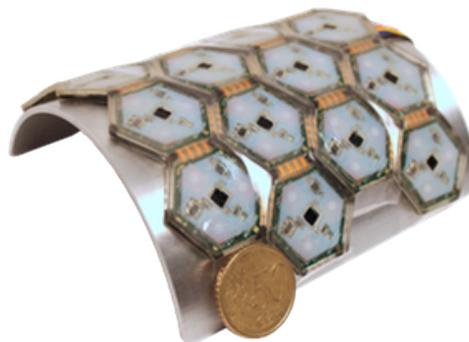
Industrie, den gemeinsamen Arbeitsbereich zu schützen, ist die zeitlich getrennte Nutzung. Dabei wird durch sogenannte trennende Schutzeinrichtungen entweder die An- oder Abwesenheit des Menschen in einem definierten Bereich festgestellt. Durch diese Technik kann der Roboter angehalten werden, wenn der Mensch in den Bereich eindringt beziehungsweise der Roboter wieder angefahren werden, wenn der Mensch den Bereich verlässt. Ein typisches Beispiel aus der Industrie dazu ist, dass der Mensch einen Bestätigungsknopf außerhalb des geteilten Bereichs drücken muss. Eine Lichtschranke wird damit aktiv geschaltet und sichert den verlassenen Bereich ab. Sobald ein Objekt die Lichtschranke passiert, löst diese ein Signal aus, das einen Not-Halt beim Roboter veranlasst.



**Abbildung 2.3:** Kombinationen aus zeitlicher und örtlicher Trennung [30]



(a)



(b)

**Abbildung 2.4: Forschungsansätze für den Einsatz von Robotern in der Nähe des Menschen** - (a) Handling-Assistent von Festo [32] (b) Künstliche Haut HEX-o-SKIN [35]

Für eine echte Kooperation zwischen Mensch und Roboter sind diese bisherigen Lösungen noch nicht zufriedenstellend. Deshalb gibt es hier bereits Forschungsansätze, wie Mensch und Roboter nebeneinander ohne trennende Schutzeinrichtungen arbeiten können. Eine Möglichkeit ist zum Beispiel, den Roboter nachgiebig zu entwickeln. Ein Beispiel hierfür ist der mit dem deutschen Zukunftspreis 2010 ausgezeichnete Handling-Assistent der Firma Festo [32] aus Abbildung 2.4 (a). Dieser ist einem Elefantenrüssel nachempfunden und bietet strukturelle Sicherheit für einen ungefährlichen Einsatz in der Nähe des Menschen. Eine andere Möglichkeit ist es, den Roboter mit Sensorik auszustatten, beispielsweise mit einer künstlichen Haut [33] (siehe Abbildung 2.4 (b)), die Kontakt feststellt und daraufhin den Roboter stoppt [34].

Zusätzlich gibt es die Option, den Arbeitsraum mit Sensorik zu überwachen und durch intelligente Pfadplanung eine Kollision zu vermeiden. Wenn eine Kollision nicht vermeidbar ist, bleibt auch in diesem Fall der Roboter stehen [36].

Auch bei den Industrienormen für den Einsatz von Robotern in der Nähe von Menschen sind Entwicklungen hin zu mehr Zusammenarbeit feststellbar [37, 38]. Allerdings müssen für diese neue Art von Robotern noch einige rechtliche Fragen [39] in diesem Zusammenhang geklärt werden, wenn solche intelligenteren oder vollautonomen Systeme in der Industrie oder gar im Haushalt in der Nähe von Menschen eingesetzt werden sollen. Bisher noch ungeklärte Fragen in diesem Bereich sind beispielsweise, wer bei Unfällen mit solchen Systemen haftet oder unter welchen Voraussetzungen diese assistiven Systeme überhaupt in Betrieb genommen werden dürfen.

## **2.3 Vision: Roboter als Assistent in einer industriellen Umgebung**

In diesem Abschnitt wird die Idee eines Assistenzroboters in einem industriellen Umfeld präsentiert sowie eine kurze Beschreibung der beiden Projekte, in deren Rahmen diese Arbeit entstanden ist. Des Weiteren werden einige verwandte Projekte zu diesem Themenbereich vorgestellt.

Die Entwicklung und Technologie für die Servicerobotik und deren Anwendungen im Haushalt bieten auch neue Perspektiven für die Anwendung von Robotern in Bereichen der Industrie, die bisher nicht von der Automatisierungstechnik profitieren konnten. Ein Beispiel für diese Entwicklung sind Roboter, die den Menschen bei einem industriellen Fertigungsprozess unterstützen. Dabei kann der Roboter seine Stärken in das Team einbringen und idealerweise den Menschen bei unangenehmen Tätigkeiten entlasten. Roboter können beispielsweise mittlere bis schwere Lasten heben und dies sogar 24 Stunden am Tag und sieben Tage die Woche, mit einer hohen Wiederholungsrate. Außerdem können sie gefährliche oder giftige Gegenstände handhaben und erledigen ihre Aufgaben mit sehr großer Präzision. Der Mensch hingegen ist - zumindest bisher - unschlagbar flexibel bei der Erledigung seiner Arbeit. Insbesondere bei Überraschungen und Fehlern im Prozessablauf kann der Mensch seine Erfahrung und seine Problemlösungskompetenz voll einbringen, um das System wieder in einen funktionsfähigen Zustand zu überführen.

Der Einsatz eines Roboters als Assistent, welcher den Menschen bei einer Aufgabe in einer industriellen Umgebung unterstützen beziehungsweise entlasten soll, bringt auch neue Anforderungen an den Roboter mit sich. Dies ist begründet in der Tatsache, dass der Roboter seine Aufgabe eben nicht mehr isoliert in einem räumlich oder zeitlich getrennten Arbeitsbereich verrichtet, sondern gemeinsam mit einem Menschen ein Ziel erreichen soll. Deshalb sollten einige Aspekte von Robotern, die besonders für den sozialen Einsatz entwickelt wurden, auch auf diese assistierenden Roboter in der Industrie übertragen werden. Eine sehr gute Übersicht über interaktive Roboter im sozialen Anwendungsbereich findet sich in [40].

Hierbei werden unter anderem Design und Gestaltung, Emotion, Dialog, Persönlichkeit, menschenbezogene Wahrnehmung und Intention thematisiert. Auch wenn die Emotionen und Persönlichkeit eines Roboters in der Industrie wohl eher eine untergeordnete Rolle spielen dürften, sind die anderen Aspekte durchaus relevant für einen Roboter, der den Menschen unterstützt und dessen Arbeitsbereich teilt. Natürlich steht bei Design und Gestaltung in der Industrie primär die gewünschte Funktionalität im

## 2. MENSCH UND ROBOTER

---

Vordergrund, doch bereits in dieser Phase sollte an den späteren Einsatz zusammen mit dem Menschen gedacht werden. Für eine sichere Interaktion können scharfe Kanten und Klemmmöglichkeiten vermieden sowie die möglichen Kräfteinwirkungen reduziert oder direkt Nachgiebigkeit in die Struktur des Roboters integriert werden.

Der Dialog zwischen Mensch und Roboter kann beispielsweise für die Programmierung oder Änderung von Abläufen hilfreich sein. Die Wahrnehmung des Menschen ist im Assistenzszenario wichtig, um den Arbeitsablauf zu synchronisieren und effizienter zu gestalten und um die Interaktion im Allgemeinen zu verbessern. Hierbei spielt auch die Intention eine Rolle. Das System sollte idealerweise erkennen, wohin der Mensch seine momentane Aufmerksamkeit richtet, jedoch auch seine eigenen nächsten Schritte dem Menschen erkennbar signalisieren, damit dieser sich darauf einstellen kann und nicht von Aktionen des Systems überrascht wird.



**Abbildung 2.5:** Assistenzroboter Baxter von Rethink Robotics [41]

Ein Repräsentant dieser Entwicklung von Robotern ist beispielsweise Baxter [41] von Rethink Robotics. Wie in Abbildung 2.5 zu sehen ist, hat dieser Roboter ein Display für die Darstellung menschlicher Gesichtszüge und kann dieses auch auf seinen momentanen Aufmerksamkeitsfokus ausrichten. Der Roboter ist von Grund auf für den Einsatz zusammen mit dem Menschen konzipiert worden und hat Federn und Kraftsensoren in jedem Gelenk. Deshalb kann der Roboter auch ohne trennende Schutzeinrichtung verwendet werden. Mittels Sonarsensoren und einer Kamera nimmt der Roboter auch Menschen in seiner Umgebung wahr. Außerdem hat der Roboter zwei Manipulatoren mit je sieben Freiheitsgraden und verfügt über eine bildbasierte Objekterkenn-

ung. Die Programmierung des Roboters kann auch durch direktes Führen der Arme geschehen. Mittels der Navigationsschalter an den Armen kann der Benutzer durch ein grafisches Menü steuern und weitere Einstellungen vornehmen.

Laut Herstellerangaben kann der Roboter für Materialhandhabungsprozesse, einfache Montagetätigkeiten, Test- und Sortierungsprozesse, Ver- und Entpacken sowie für die Bestückung von Fertigungslinien eingesetzt werden. Die Tragkraft der beiden Arme ist mit jeweils ungefähr 2,3 kg sicher einschränkend für einige Tätigkeiten in der Industrie. Dennoch stellt dieser Assistenzroboter einen weiteren bedeutsamen Schritt für den Einsatz von Servicerobotern in industriellen Umgebungen dar. Auch der Einstieg von Google in die Servicerobotik zeigt die Relevanz dieses Bereichs. Des Weiteren wird sich dadurch auch die Entwicklung der Produkte beschleunigen [42].

Die folgenden Forschungsprojekte haben sich das Ziel gesetzt, durch Mensch-Roboter-Kooperation den Roboter im industriellen Kontext gemeinsam mit dem Menschen an Aufgaben arbeiten zu lassen. Schwerpunkte sind hierbei unter anderem die sichere Interaktion sowie auch eine natürlichere Bedienbarkeit und Zusammenarbeit mit dem Roboter.

### 2.3.1 JAHIR

Das erste Projekt, das im Rahmen der Dissertation bearbeitet wurde, ist das Projekt Joint Action for Humans and Industrial Robots (JAHIR). Die Deutsche Forschungsgemeinschaft (DFG) förderte das Projekt im Rahmen des Exzellenzforschungsclusters Cognition for Technical Systems (CoTeSys). JAHIR beschäftigt sich mit unterschiedlichen Aspekten der Mensch-Roboter-Kooperation (MRK), wie beispielsweise der Vermeidung von Kollisionen bei der Aufhebung der trennenden Schutzeinrichtung zwischen Mensch und Roboter durch dynamische Pfadplanung basierend auf Potentialfeldern [43] für konventionelle Roboter sowie mit einer natürlichen Interaktion zwischen Mensch und Roboter. Das Projekt diente auch als Integrationsplattform für weitere Forschungsarbeiten. Die Mensch-Roboter-Interaktion wurde bezüglich einer natürlicheren Bewegung des Roboters bei der Übergabe von Objekten untersucht [44]. In einer weiteren Studie [45] wurden Emotionen wie Stress, Langeweile und Überraschung im Rahmen der Mensch-Roboter-Kooperation beim Werker erzeugt und gemessen.

### 2.3.2 CustomPacker

Das Projekt mit dem Titel Highly Customizable and Flexible Packaging Station for mid- to upper sized Electronic Consumer Goods using Industrial Robots (Custom-

Packer) ist das zweite im Rahmen dieser Dissertation behandelte Projekt und wurde von der EU im siebten Rahmenprogramm gefördert. In dem Projekt geht es um die Entwicklung neuer Konzepte und Komponenten für Verpackprozesse. Hierbei liegt der Fokus auf der Verpackung von elektronischen Konsumgütern. Im Rahmen dieses Forschungsprojekts wurden unter anderem Komponenten zur Objekterkennung, Werkerpositionsüberwachung, Assistenz und zur vereinfachten Programmierung der Abläufe in der Verpackstation entwickelt. Des Weiteren wurde ein flexibler Greifer für die Handhabung der mittleren bis schweren Lasten sowie ein nachgiebiger Roboter für den Einsatz in der Nähe des Menschen entwickelt, der trotzdem die notwendigen Kräfte und die erforderliche Steifigkeit zur Handhabung der Lasten hat. Die entwickelten Komponenten können je nach Anforderung zu einer Verpackstation kombiniert werden. Im Rahmen des Projekts wird dies exemplarisch anhand eines Verpackszenarios der Firma Loewe AG umgesetzt.

### 2.3.3 Projekte mit Assistenzrobotern im industriellen Umfeld

Es gibt noch viele weitere Projekte, die sich mit dieser Thematik beschäftigen. Im Folgenden werden einige relevante Projekte in alphabetischer Reihenfolge stellvertretend erwähnt:

#### COGNIRON

Das von der EU im sechsten Rahmenprogramm geförderte Projekt *COGNIRON - The Cognitive Robot Companion* [46] hat es sich zur Aufgabe gemacht, die Wahrnehmung, Repräsentation, Lernfähigkeiten sowie das logische Denken eines Roboters in einer auf den Menschen fokussierten Umgebung zu untersuchen. Dabei sollen unter anderem grundlegende Methoden, Algorithmen und Architekturen entwickelt werden, die dann in Roboter integriert und für wissenschaftliche Studien genutzt werden sollen.

#### EsIMiP

Das Projekt mit dem Titel *EsIMiP - Efficient and safe interaction of humans and intelligent production systems* [47] versucht durch experimentelle Studien die Akzeptanz der Nutzer eines gesamten Systems zur Mensch-Roboter-Interaktion sicherzustellen. Hierbei wird ein besonderes Augenmerk auf ergonomische Aspekte gerichtet und versucht, die Bewegungen der beiden Akteure in der Zelle effizient und harmonisch zu gestalten.

### **INTROP**

Das Projekt *INTROP - Intuitive Programmierung von Roboter-Manipulatoren* [48] ist ein von der DFG gefördertes Projekt mit dem Fokus auf der Programmierung von Roboter-Manipulatoren durch Personen, die keine Programmierkenntnisse besitzen. Das Projekt wird vom Lehrstuhl für Robotik und Eingebettete Systeme der Universität Bayreuth geleitet. Mithilfe von Sensoren soll der Mensch bei der Interaktion mit dem Roboter überwacht werden und der Roboter dadurch die notwendigen Arbeitsschritte lernen. Die Anwendung der intuitiv programmierten Roboter soll sowohl im privaten als auch im beruflichen Umfeld stattfinden.

### **LiSA**

Das Projekt *LiSA - Assistenzroboter in Laboren von Life-Science-Unternehmen* [49] ist ein vom BMBF gefördertes Projekt und entwickelt einen mobilen und alltagstauglichen Assistenzroboter. Dieser soll in Laboren von Unternehmen auf dem Gebiet Life-Science erprobt werden. Dabei soll die Plattform mit Mitarbeitern interagieren und wiederkehrende Aufgaben wie Transporte inklusive eigener Bestückung übernehmen.

### **LoCoBot**

Das Projekt *LoCoBot - Low Cost Robot Co-Worker* [50] versucht einen günstigen Roboterbausatz zu entwickeln, der auf modularen Roboterkomponenten sowie Sensoren für die Mensch-Roboter-Interaktion beruht. Die drei Hauptziele sind dabei eine modulare Plug-and-Produce-Roboterplattform, eine rekonfigurierbare und adaptierbare Steuerung für Plug-and-Produce-Komponenten sowie intelligente, wahrnehmende und agierende Module.

### **PiSA**

Bei *PiSA - Flexible Assembly Systems through Workplace-Sharing and Time-Sharing Human-Machine Cooperation* [51] besteht das Ziel darin, ein flexibles Montage-System durch räumlich und zeitlich getrennte Mensch-Maschine-Kooperation zu entwickeln. Dieses Projekt wurde durch die EU im sechsten Rahmenprogramm gefördert. In diesem Forschungsprojekt sollen die Stärken von Mensch und Maschine kombiniert werden, um komplexe Produkte rasch und kostengünstig fertigen zu können.

## 2. MENSCH UND ROBOTER

---

### **PRACE**

*PRACE - The Productive Robot Apprentice* [52] wird im siebten Rahmenprogramm der EU gefördert. Ziel des Projekts ist eine schnelle und wirtschaftliche Automatisierung manueller Arbeitsplätze in Werkstattumgebungen. Hierbei soll der Mensch dem Roboter den Prozess durch Demonstration beibringen und später für die Überwachung der Produktion eingebunden werden. Dadurch soll die Produktion von Hochtechnologieprodukten in Europa erhalten bleiben und sogar aus Billiglohnländern zurückgeholt werden.

### **SMErobotics**

Das Projekt *SMErobotics - Small and Medium Enterprise Robotics* [53] wird im siebten Rahmenprogramm der EU gefördert und beschäftigt sich mit der Frage, wie Roboter auch in kleinen und mittleren Unternehmen produktiv für Automatisierungstätigkeiten eingesetzt werden können. Motivation ist hierbei, dass über zwei Drittel der europäischen Arbeiter in der Fertigungsbranche in kleinen oder mittleren Unternehmen beschäftigt sind. Diese Unternehmen müssen besonders flexibel auf Änderungen in der Produktion reagieren können und eine sehr hohe Produktqualität beibehalten, um auf dem Markt bestehen zu können. In das Konsortium sind neben führenden Forschungseinrichtungen auch Roboterhersteller eingebunden. Mittels verschiedener Demonstratoren bei kleinen und mittleren Unternehmen soll die Umsetzbarkeit der Konzepte gezeigt werden. In Abbildung 2.6 ist der Leichtbaurobter intelligent industrial work assistant (LBR iiwa) [54] von KUKA in einem Demonstrator vom Fraunhofer IPA zu sehen, der einen Werker bei einer Fertigung unterstützt.



**Abbildung 2.6:** Sensorgesteuerte Fertigung mit dem Fertigungsassistent [55]

# Kapitel 3

## Theoretische und methodische Grundlagen

Dieses Kapitel beginnt mit einer Einführung theoretischer Grundlagen und verwendeter mathematischer Verfahren. Des Weiteren wird ein Überblick über Methoden zur Evaluierung von Interaktionen präsentiert. Außerdem werden die zur Interaktion zur Verfügung stehenden Benutzerschnittstellen aufgeführt und die zur Implementierung der Entwicklungen in dieser Arbeit verwendeten Softwarepakete vorgestellt.

### 3.1 Theoretische Grundlagen

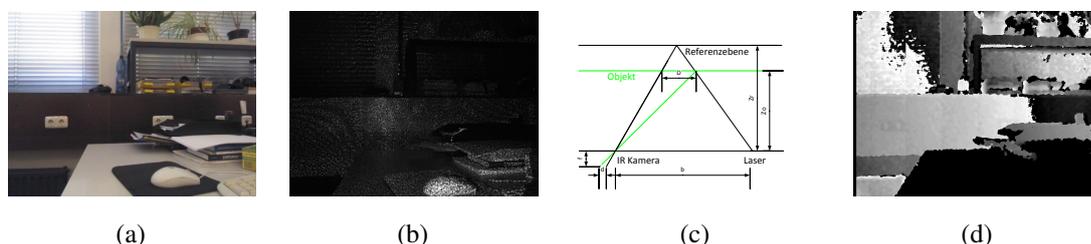
Einer der in dieser Arbeit betrachteten Themenbereiche zur Verbesserung der Interaktion mit Fokus auf die Mensch-Maschine-Schnittstelle sind Lernmethoden in der Mensch-Roboter-Kooperation, die in Kapitel 6 näher beschrieben werden. Für eine intuitive Methode zur Programmierung eines Roboters werden dafür Aktionen des Menschen mithilfe von Punktwolken analysiert. Deshalb wird an dieser Stelle für ein besseres Verständnis eine kurze Einführung in diese Punktwolken sowie die für die Analyse verwendeten Merkmalsdeskriptoren vorgestellt.

#### 3.1.1 Punktwolken und verwendete Merkmalsdeskriptoren

Punktwolken sind in dieser Arbeit definiert als eine Menge  $\mathcal{P}$ , deren Elemente Punkte im dreidimensionalen Raum mit den Koordinaten  $\langle x, y, z \rangle$  darstellen. Zusätzlich

### 3. THEORETISCHE UND METHODISCHE GRUNDLAGEN

---



**Abbildung 3.1: Erzeugung des Tiefenbildes mithilfe eines projizierten Musters** - (a) Szene im RGB-Farbraum (b) Infrarotbild des projizierten Musters (c) Berechnung der Tiefeninformation [59] (d) Gewonnenes Tiefenbild

zu den kartesischen Koordinaten können diese Elemente weitere Informationen enthalten, wie beispielsweise Farbinformationen  $\langle r, g, b \rangle$  (auch RGB-D genannt für Rot-Grün-Blau-Farbbilder mit Tiefeninformation (engl. depth)). Da der Sensor mehr als die zweidimensionalen Daten eines Bildes, jedoch keine kompletten dreidimensionalen Daten liefert, werden diese Daten auch als zweieinhalbdimensionale Daten bezeichnet. In dieser Arbeit wurden die verwendeten Punktwolken durch den Microsoft Kinect-Sensor generiert. Auch wenn der Sensor ursprünglich für Unterhaltungs- und Spieleanwendungen entwickelt wurde, wird der Einsatz des Sensors in industriellen Umgebungen zunehmend interessant. In einer Studie [56] des Logistikdienstleisters DHL zusammen mit dem Fraunhofer Institut für Fabrikbetrieb und -automatisierung (IFF) wird der Sensor zur Volumenüberwachung von Paletten in der Logistik angewendet. Bei Siemens [57] wird der Sensor zur Positionserkennung eingesetzt. Neben ergonomischen Untersuchungen soll durch die Analyse der Bewegungstrajektorien der Aufenthalt in chemisch oder radioaktiv belasteten Räumen für den realen Einsatz minimiert werden. Wie in der Patentschrift [58] der Firma PrimeSense Ltd. beschrieben, wird mittels einer Lichtquelle und eines Filters ein definiertes Muster im Infrarotbereich generiert, das beispielhaft zusammen mit dem RGB-Farbbild und dem Prinzip der Berechnung der Tiefeninformation laut [59] in Abbildung 3.1 dargestellt ist. Der Sensor hat das Referenzmuster mit dem bekannten Abstand  $Z_r$  für jeden Musterpunkt gespeichert. Das vom Laser projizierte Muster wird mittels der Infrarotkamera erfasst. Die so erfasste Position der Punkte im Bild der Infrarotkamera kann mit der Referenzaufnahme verglichen und schließlich kann mittels der Gleichungen 3.1 und 3.2 aus [59] die Tiefeninformation  $Z_o$  zum Objekt bestimmt werden.

$$\frac{D}{b} = \frac{Z_r - Z_o}{Z_r} \quad (3.1)$$

und

$$\frac{d}{f} = \frac{D}{Z_o} \quad (3.2)$$

Durch Kombinieren der beiden Gleichungen ergibt sich die gesuchte Tiefe  $Z_o$  für einen Punkt zu:

$$Z_o = \frac{Z_r}{1 + Z_r \frac{d}{f \cdot b}} \quad (3.3)$$

Aus dem Tiefenbild mit einer Auflösung von  $x = 320$  und  $y = 240$  und der in den Pixelwerten gespeicherten Tiefeninformation  $Z_o$ , die beim Kinect mit 11 Bit codiert ist, sowie der Kamerageometrie kann dann die Punktwolke  $\mathcal{P}$  berechnet werden. Der Sensor liefert die Daten mit 30 Hz.

### Schnelles Punktmerkmalshistogramm

Das schnelle Punktmerkmalshistogramm (engl.: Fast Point Feature Histogram (FPFH)) ist eine schneller zu berechnende Variante des Point Feature Histogram (PFH) [60]. Dabei werden drei Winkelmerkmale  $\langle \alpha, \phi, \theta \rangle$  mittels der Gleichung 3.4 aus [60] zwischen den beiden Flächennormalen  $\mathbf{n}_t$  und  $\mathbf{n}_s$  in den Punkten  $\mathbf{p}_t$  und  $\mathbf{p}_s$  berechnet, wie in Abbildung 3.2 dargestellt.

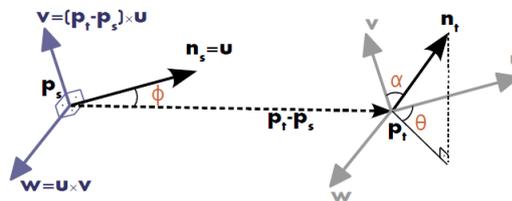


Abbildung 3.2: Winkelmerkmalsberechnung für das PFH, entnommen aus [60].

$$\begin{aligned} \alpha &= \mathbf{v} \cdot \mathbf{n}_t \\ \phi &= \mathbf{u} \cdot \frac{(\mathbf{p}_t - \mathbf{p}_s)}{d} \\ \theta &= \arctan(\mathbf{w} \cdot \mathbf{n}_t, \mathbf{u} \cdot \mathbf{n}_t) \end{aligned} \quad (3.4)$$

Dabei ist  $\mathbf{u} = \mathbf{n}_s$ ,  $\mathbf{v} = (\mathbf{p}_t - \mathbf{p}_s) \times \mathbf{u}$ ,  $\mathbf{w} = \mathbf{u} \times \mathbf{v}$ ,  $d = \|\mathbf{p}_t - \mathbf{p}_s\|_2$  die euklidische Distanz zwischen den Punkten  $\mathbf{p}_t$  und  $\mathbf{p}_s$ . Für einen Punkt  $\mathbf{p}$  werden diese Merkmale zwischen sich und seinen nächsten  $k$  Punkten im Umfeld mit Radius  $r$  berechnet. Der mögliche Wertebereich der Winkelmerkmale wird diskretisiert und die in den Bereichen auftretenden Winkel werden gezählt. Das so gewonnene Merkmalshistogramm

### 3. THEORETISCHE UND METHODISCHE GRUNDLAGEN

---

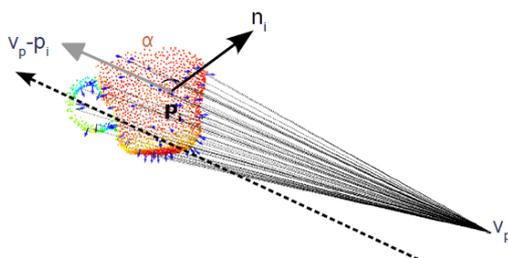
wird als Simplified Point Feature Histogram (SPFH) bezeichnet. Das FPFH wird nun mittels der Gleichung 3.5 aus [60] berechnet:

$$FPFH(\mathbf{p}_q) = SPFH(\mathbf{p}_q) + \frac{1}{k} \sum_{i=1}^k \frac{1}{w_k} \cdot SPFH(\mathbf{p}_k) \quad (3.5)$$

Das FPFH für einen Punkt  $\mathbf{p}_q$  setzt sich also aus dem SPFH für  $\mathbf{p}_q$  sowie einer Summe der SPFH seiner umliegenden  $k$  Nachbarpunkte zusammen. Dabei ist  $w_k$  die Distanz zwischen Punkt  $\mathbf{p}_q$  und  $\mathbf{p}_k$  in einer Metrik.

#### Blickpunktmerkmalshistogramm

Beim Blickpunktmerkmalshistogramm (engl.: Viewpoint Feature Histogram (VFH)) [61] wird ein blickpunktabhängiges Winkelmerkmal zum Histogramm hinzugefügt. Dadurch ist neben der Objekterkennung auch die Bestimmung der Orientierung des Objekts möglich.



**Abbildung 3.3:** Blickpunktabhängiges Winkelmerkmal für das VFH [61]

Für die Blickwinkelabhängigkeit wird der Winkel  $\alpha$  aus Abbildung 3.3 zwischen der in den jeweiligen Punkt  $\mathbf{p}_i$  verschobenen zentralen Blickrichtungskomponente (gestrichelte Linie) und der Flächennormale  $\mathbf{n}_i$  im Punkt  $\mathbf{p}_i$  berechnet. Zusätzlich werden die drei Winkelmerkmale wie zuvor beim FPFH berechnet. Diesmal jedoch zwischen der Flächennormale im Schwerpunkt  $\mathbf{c}$  des Objekts und seiner umliegenden Punkte, wie in Abbildung 3.4 dargestellt.

Es ergeben sich 308 Werte für das VFH, wobei sich diese aus je 45 Behältern für drei Winkelmerkmale sowie der Distanz zum Schwerpunkt ( $4 \cdot 45 = 180$ ) plus weiteren 128 Behältern für die vom Blickwinkel abhängige Komponente zusammensetzen.

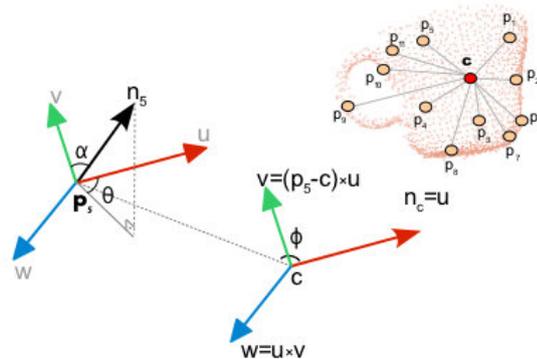


Abbildung 3.4: FPFH Komponente für den Schwerpunkt c [61]

### 3.1.2 Markov-Ketten

Markov-Ketten werden genutzt, um Prozesse mit zufälligen Zustandsübergängen zu modellieren. Aufgrund der Zufälligkeit des Ablaufs spricht man dabei von stochastischen Prozessen, im Gegensatz zu deterministischen Prozessen, bei denen der Ablauf eindeutig festgelegt ist. Der zugrunde liegende stochastische Prozess heißt in diesem Zusammenhang auch Markov-Prozess und die emittierende stochastische Quelle des Prozesses wird als Markov-Quelle bezeichnet.

Die Besonderheit bei Markov-Prozessen liegt darin, dass der Übergang von einem Zustand zum nächsten nur von  $g$  vorherigen Zustandsübergängen abhängt.

Mathematisch ausgedrückt bedeutet dies, dass die Wahrscheinlichkeit, sich zur Zeit  $t + 1$  in einem Zustand  $z_{t+1}$  zu befinden, folgendermaßen äquivalent umgeformt werden kann:

$$\begin{aligned} p(X_{t+1} = z_{t+1} | X_t = z_t, \dots, X_0 = z_0) = \\ p(X_{t+1} = z_{t+1} | X_t = z_t, \dots, X_{t-g+1} = z_{t-g+1}), \end{aligned} \quad (3.6)$$

wobei  $X$  eine Zufallsvariable ist und  $X_t$  den Zustand  $z_t$  zum Zeitpunkt  $t$  angibt. Sowie  $z_t \in Z$  der Menge aller möglichen Zustände ist.  $g$  wird im Zusammenhang mit Markov-Ketten auch als Ordnung der Markov-Kette bezeichnet. Falls nur die Ordnung  $g = 1$  betrachtet wird, lässt sich die Gleichung 3.6 mittels folgender Gleichung vereinfachen:

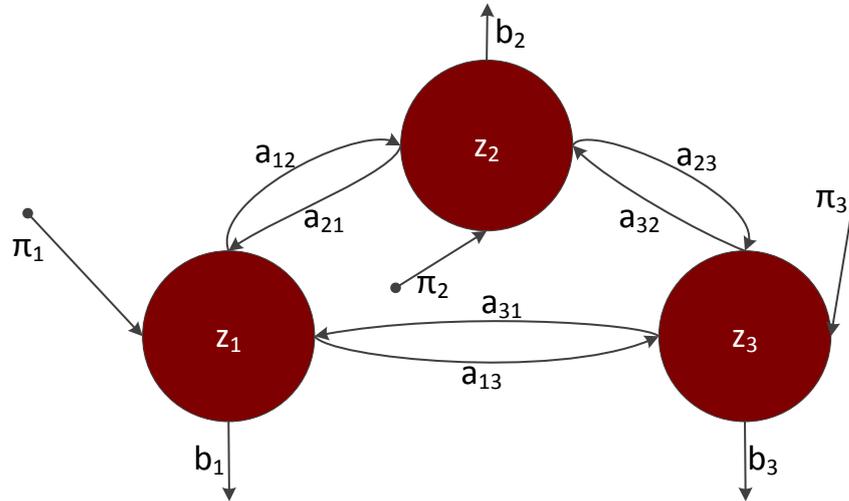
$$\begin{aligned} p(X_{t+1} = z_{t+1} | X_t = z_t, \dots, X_{t-g+1} = z_{t-g+1}) = \\ p(X_{t+1} = z_{t+1} | X_t = z_t) \end{aligned} \quad (3.7)$$

Dies bedeutet, dass der nächste Zustand nur vom aktuellen abhängt, was für viele praktische Anwendungen eine ausreichende Modellierung ist und aus diesem Grund häufig Anwendung findet. Eine Markov-Kette erster Ordnung ist dann definiert durch die Menge der Zustände  $Z = \{z_1, \dots, z_N\}$ , die Startverteilung  $\pi$  zum Zeitpunkt  $t = 0$ , mit  $\pi_i = p(X_{t=0} = z_i)$  mit  $1 \leq i \leq N$  und der  $N \times N$  Transitionsmatrix  $A$ , wobei das Matrixelement  $a_{ij}$  angibt, mit welcher Wahrscheinlichkeit  $p(X_{t+1} = z_j | X_t = z_i)$  im nächsten Zeitschritt vom Zustand  $z_i$  in den Zustand  $z_j$  gewechselt wird.

#### 3.1.3 Verborgene Markov-Modelle

Die verborgenen Markov-Modelle (engl.: Hidden Markov Model (HMM)) finden häufig in der Spracherkennung [62] Anwendung und können als stochastische Erweiterung von Zustandsautomaten verstanden werden.

Aufgrund der stochastischen Zustandstransitionen eignen sich die Modelle sehr gut für die Modellierung dynamischer Sequenzen, weshalb sich das Anwendungsfeld für diese stochastischen Modelle rasant vergrößert hat. Inzwischen finden sie unter anderem in der Handschrifterkennung auf Weißwandtafeln [63] sowie in der Gesichtserkennung [64], der Gestenerkennung [65] oder auch in der Robotik [66, 67] Anwendung. Der Name HMM kommt aus dem Englischen und bezeichnet ein stochastisches Modell, das auf zwei stochastischen Prozessen beruht, wobei der eine die Zustandsübergänge und der andere die Wahrscheinlichkeiten der Beobachtungen in den jeweiligen Zuständen modelliert. Bei HMMs sind allerdings genau diese Zustände der Markov-Kette (siehe Abschnitt 3.1.2) nicht direkt beobachtbar und werden deshalb als verborgen bezeichnet. Mathematisch ist ein HMM  $\lambda$  als 3-Tupel  $\lambda = (A, B, \pi)$  definiert. Die Transitionsmatrix  $A$  codiert die Wahrscheinlichkeiten für die Zustandsübergänge des ersten stochastischen Prozesses und hat bei  $N$  Zuständen die Dimension  $N \times N$ . Die  $B$ -Matrix enthält folglich die Information für den zweiten stochastischen Prozess, nämlich die Emissionswahrscheinlichkeiten für die  $M$  möglichen Beobachtungen für jeden der  $N$  Zustände. Die initiale Einsprungswahrscheinlichkeit, sich zum Zeitpunkt  $t = 0$  in einem der  $N$  Zustände zu befinden, wird im Vektor  $\pi$  hinterlegt. Ein ergodisches HMM mit drei versteckten Zuständen, den Einsprungswahrscheinlichkeiten, den Transitionen sowie den Emissionsvektoren ist in Abbildung 3.5 dargestellt.



**Abbildung 3.5:** HMM mit  $N = 3$  versteckten Zuständen

Für die Klassifikation einer Beobachtungssequenz  $O = \{O_0, \dots, O_{l-1}\}$  mit  $l$  Zeitschritten wird mit zuvor durch den Baum-Welch-Algorithmus (spezielle Variante des EM-Algorithmus) trainierten Modellen  $\lambda_c$  für die  $C$  Klassen über das Modell mit der höchsten Produktionswahrscheinlichkeit  $p(O|\lambda_c)$  die gesuchte Klasse  $c_*$  bestimmt:

$$c_* = \arg \max_c p(O|\lambda_c) \quad (3.8)$$

und

$$p(O|\lambda_c) = \sum_{\mathbf{h} \in H} p(O, \mathbf{h}|\lambda_c), \quad (3.9)$$

wobei  $\mathbf{h} = (h_1, \dots, h_l)$  eine aller möglichen Zustandsfolgen aus der Menge  $H$  ist und  $h_i \in Z$ .

Mittels des Viterbi-Algorithmus [68] kann die wahrscheinlichste Folge der verborgenen Zustände berechnet werden.

### 3.1.4 Zufällige Stichproben Konsens-Algorithmus

Der Random Sample Consensus (RANSAC)-Algorithmus [69] hilft bei der Parameterwahl von überbestimmten Modellen, die Ausreißer enthalten. Im Fall dieser Arbeit können dies beispielsweise fehlerhaft berechnete Abstandswerte in der Punktwolke sein. Wenn das Modell nicht überbestimmt ist, können keine Messpunkte als Ausreißer aussortiert werden, da alle Werte zur Modellberechnung benötigt werden. Ansonsten können jedoch zufällige Punkte für das Modell gewählt werden, was auch als Stichprobe (engl. Sample) bezeichnet wird. Mit dieser Stichprobe wird zunächst das Modell konstruiert. Anhand des Modells und eines zu wählenden Abstandsgrenzwertes  $d$  (meist empirisch ermittelt) kann dann der Rest der Punkte als ins Modell passend beziehungsweise als Ausreißer klassifiziert werden. Dies kann nun bis zum Erreichen einer gewünschten Erfolgswahrscheinlichkeit  $p$  wiederholt werden. Der Erfolg ist hierbei dahingehend definiert, keine Ausreißer in der Stichprobe zu haben. Aus diesem Grund gehört der RANSAC-Algorithmus auch zur Familie der Monte-Carlo-Algorithmen mit einseitiger Fehlerschranke  $1 - p$ .

Falls nun  $\varepsilon$  die Wahrscheinlichkeit für eine zufällig gewählte Stichprobe ist, die ein schlechtes Modell produziert (also Ausreißer in den Modellparametern hat), dann ist  $1 - \varepsilon$  die Wahrscheinlichkeit, zumindest eine gute Stichprobe auszuwählen.

Die Wahrscheinlichkeit für das Auswählen von  $s$  guten Stichproben wird folglich zu  $(1 - \varepsilon)^s$ . Die Wahrscheinlichkeit für einen Misserfolg wird für  $k$  Versuche zu  $1 - (1 - \varepsilon)^s)^k$ . Die Anzahl der Iterationen  $k$  für eine gewünschte Erfolgswahrscheinlichkeit  $p$  (beispielsweise  $p = 0,99$ ) ergibt sich dann aus:

$$1 - p = (1 - (1 - \varepsilon)^s)^k \quad \Rightarrow \quad k = \frac{\log(1 - p)}{\log(1 - (1 - \varepsilon)^s)} \quad (3.10)$$

Ausgedrückt als Minimierungsproblem einer Kostenfunktion gilt laut [70] für die Kostenfunktion:

$$f_r = \sum_i^N \rho_r(e_i^2), \quad \rho_r(e^2) = \begin{cases} 0, & e^2 < d^2 \\ c, & e^2 \geq d^2 \end{cases}, \quad (3.11)$$

wobei  $e_i$  die Distanz vom Messwert  $i$  zum betrachteten Modell mit  $N$  weiteren Messwerten angibt. Die Konstante  $c$  wird meist mit dem Wert 1 belegt, wodurch die Kostenfunktion  $f_r$  die Anzahl der Ausreißer bestimmt.

## 3.2 Benutzerschnittstellen

Ein Überblick über wichtige Grundbegriffe der Mensch-Maschine-Kommunikation ist in [71] zu finden. Wie der Name der Forschungsdisziplin bereits impliziert, muss sowohl der Mensch als auch die Maschine bei der Entwicklung und Konzeption der Kommunikation berücksichtigt werden. Der Fokus dieser Entwicklung liegt auf der Optimierung des Informationsflusses zwischen Mensch und Maschine, wobei hierfür neue Geräte, Algorithmen und die unterschiedlichen Interaktionskanäle untersucht werden. Dabei spielt die Benutzerschnittstelle eine zentrale Rolle. Hier müssen die aktuellen Zustände der Maschine dem Menschen möglichst übersichtlich und klar dargeboten werden. Außerdem müssen über diese Schnittstelle die Zustände der Maschine mittels Einstellungen und Funktionsaufrufen einfach veränderbar sein. Im Laufe der Zeit wurden dabei unter anderem folgende Benutzerschnittstellen zur Interaktion des Benutzers mit dem System entwickelt.

### 3.2.1 Kommandozeilen-Benutzerschnittstelle

Die Kommandozeilen-Benutzerschnittstellen (engl.: Command Line Interface (CLI)) erlauben die Eingabe von Informationen in Form von Zeichen und dienen zur Eingabe von Befehlen. Ebenso erfolgt die Rückmeldung an den Benutzer über das Ergebnis der Befehle nur in Form von textueller Information. Der Roboter Shakey [72] wurde beispielsweise über diese Art der Benutzerschnittstelle gesteuert und konnte textbasierte Befehle wie „UNBLOCK DOOR DYMSCLK FROM ROOM RMYS.“ in Prädikatenlogik transformieren und mithilfe von List Processing (LISP) [73] weiterverarbeiten. Der Nutzer wurde mittels Nadeldrucker über die interne Zustandsänderung informiert.

### 3.2.2 Grafische Benutzerschnittstelle

Die grafischen Benutzeroberflächen (engl.: Graphical User Interface (GUI)) sind sehr weit verbreitet. Insbesondere bei Desktop-Betriebssystemen basieren diese meist auf Windows Icons, Menus, Pointer (WIMP) [74]. Auch für Robotersteuerungen ist eine grafische Benutzerschnittstelle sehr weit verbreitet. Abbildung 3.6 zeigt das Teachpanel KeTopT50 der Firma KEBA [75], deren Steuerung für den Roboter im Projekt CustomPacker eingesetzt wurde. Mittels eines grafischen Menüs können hier die Parameter des Roboters eingestellt sowie dem Nutzer die internen Systemzustände, Nachrichten und Warnungen dargestellt werden.



Abbildung 3.6: Teachpanel KeTopT50 der Firma KEBA [75]

#### 3.2.3 Greifbare Benutzerschnittstelle

Die greifbaren Benutzerschnittstellen (engl.: Tangible User Interface (TUI)) verwenden physikalische Repräsentanten für die eigentliche Manipulation eines anderen Objekts. Dabei wird die Bewegung, Orientierung und Beschleunigung dieser Repräsentanten auf das zu manipulierende Objekt übertragen. Eine sehr verbreitete greifbare Benutzerschnittstelle ist die Wiimote von Nintendo. Obwohl diese Technik besonders für die Steuerung von Objekten in virtuellen Welten eingesetzt wird, findet diese Art der Benutzerschnittstelle auch Anwendung bei der Steuerung von Robotern [76].

#### 3.2.4 Aufmerksamkeitssensitive Benutzerschnittstelle

Aufmerksamkeitssensitive Benutzerschnittstellen (engl.: Attentive User Interface (AUI)) versuchen, eine Informationsüberflutung des Nutzers zu vermeiden [77]. Hierfür werden die darzubietenden Informationen mit der Relevanz für den Nutzer priorisiert. Es wird versucht, mittels eines Modells die nächsten möglichen Schritte eines Benutzers vorherzusagen und die dafür notwendigen Informationen darzubieten. Des Weiteren wird die Aufmerksamkeit des Nutzers im Hintergrund überwacht (beispielsweise mittels Blickerfassungsalgorithmen). Dadurch können beispielsweise fokussierte Informationen aus einer größeren Menge zur besseren Lesbarkeit hervorgehoben werden. Die Aufmerksamkeitsdetektion kann für weitere Aktivitäten verwendet werden. Des

Weiteren kann die gesamte Anzeige zum Stromsparen abgeschaltet werden, wenn sie nicht betrachtet wird, wie dies beispielsweise bereits bei Mobiltelefonen umgesetzt wird. Auch längere Texte können automatisch bei Erreichen des Textendes weiterge-scrollt oder eine Spracherkennung durch Anblicken aktiviert werden, um ungewollte Befehle zu vermeiden.

### 3.2.5 Natürliche Benutzerschnittstelle

Die natürlichen Benutzerschnittstellen (engl.: Natural User Interfaces (NUI)) versuchen, eine einfache und intuitive Art der Interaktion zu etablieren. Dazu werden Eingabekanäle wie Sprache, Blick und Gesten mit grafischen Oberflächen kombiniert, um dem Nutzer eine sehr einfache Bedienung und Manipulation der dargebotenen Information zu ermöglichen. Das Ziel ist es, die Schnittstelle kontextsensitiv zu gestalten, um damit die Bedienkomplexität vom Nutzer zu nehmen, indem mithilfe von Algorithmen versucht wird, den Nutzer sowie seine Aufgaben und sein Umfeld besser wahrzunehmen und im Idealfall die Intentionen des Benutzers daraus abzuleiten. Der Nutzer kann für die Interaktion den präferierten Interaktionskanal wählen. Sehr gute Designhinweise für natürliche Benutzerschnittstellen finden sich beispielsweise im Leitfaden für die Entwicklung neuer Anwendungen für den Kinect-Sensor [78].

Der Entwicklungsaufwand für natürliche Schnittstellen ist viel größer, da unterschiedliche Eingabekanäle unterstützt und gegebenenfalls auch fusioniert werden müssen. Auch die Gefahr von Fehlern in der für diese komplexeren Benutzerschnittstellen entwickelten Software wächst damit genauso wie die Erwartungshaltung des Nutzers an die Schnittstelle, dessen Erwartungen nicht immer erfüllt werden können. Um die entwickelten Verfahren und Schnittstellen hinsichtlich ihrer Benutzbarkeit und der Bedienfreude durch die Nutzer zu bewerten, ist eine Evaluierung der Schnittstellen und Verfahren durch den Endanwender immer wichtiger und deshalb ein zentraler Aspekt in der Entwicklung der Benutzerschnittstelle.

## 3.3 Evaluierung

Die Evaluierung der Mensch-Roboter-Interaktion ist im Allgemeinen ein sehr schwieriges Unterfangen. Es gibt jedoch Bemühungen, die Evaluierung und deren Ergebnisse zu verallgemeinern und vergleichbarer zu machen - ein Beispiel hierfür ist der *Workshop on Metrics for Human-Robot-Interaction* [79]. Des Weiteren gibt es Wettbewerbe wie RoboCup@Home [80] oder RoboCup@Work [81], bei denen unterschiedliche

### 3. THEORETISCHE UND METHODISCHE GRUNDLAGEN

---

Teams Aufgaben in Haushaltsanwendungen beziehungsweise im Arbeitsumfeld lösen müssen. Hierbei ist auch die Mensch-Roboter-Interaktion relevant. Da diese Domäne ein sehr breites Forschungsfeld darstellt und deshalb unterschiedlichste Aspekte bei der Evaluierung von Interesse sein können, gibt es immer noch keinen standardisierten Benchmark für die Mensch-Roboter-Interaktion.

Für eine gute Evaluierung ist es wichtig, ein geeignetes Szenario zu finden, in dem die gewünschten Aspekte und die entwickelten Komponenten evaluiert werden können. Außerdem müssen geeignete Metriken ausgewählt werden, um die Aspekte zu bewerten. Diese Metriken können in objektive und subjektive Metriken eingeteilt werden. Typischerweise werden die objektiven Daten durch Mitprotokollieren und durch automatisch durchgeführte Berechnungen gewonnen. Beispiele hierfür sind Zeiten, um eine Aufgabe zu erledigen, Konfusions- beziehungsweise Fehlermatrizen [82], Receiver Operating Characteristic (ROC)-Kurven oder Gleichfehlerraten (engl.: Equal Error Rate (EER)) etc. Der Vorteil der objektiven Datenerhebung ist die gute Vergleichbarkeit, da die Ergebnisse nicht durch subjektive Eindrücke der Probanden beeinflusst werden. Trotzdem sollten gerade für Systeme, mit denen der Mensch in Interaktion tritt, nicht nur objektive Kriterien als alleiniges Bewertungskriterium gewählt werden. Beispielsweise kann das aus der objektiven Bewertung als schnellstes resultierende System unter ergonomischen Betrachtungen schlecht abschneiden und deshalb bei der Bedienung unkomfortabel für den Nutzer sein. Aus diesem Grund werden für die subjektive Bewertung des Systems häufig Nutzerstudien in Form von Fragebögen und Interviews durchgeführt. Bei der Bewertung durch die Probanden spielt auch der technische Hintergrund eine Rolle, weshalb sinnvollerweise auch die Erfahrung im Umgang mit technischen Geräten beziehungsweise in diesem Fall mit Robotern mit in den Fragebogen aufgenommen werden sollte.

#### 3.3.1 AttrakDiff

AttrakDiff ist ein standardisierter Fragebogen und gehört zu den subjektiven Bewertungsmethoden. Bei diesem Fragebogen werden Bewertungen hinsichtlich der hedonischen und pragmatischen Qualität eines Produkts vorgenommen. Der Nutzer beschäftigt sich hierbei zuvor mit dem zu bewertenden Produkt. Nach dem Experiment kann das Produkt vom Nutzer bewertet werden. Dazu werden mehrere Wortpaare als semantisches Differenzial aufgetragen und auf einer Skala mit sieben Einträgen von -3 über 0 bis +3 bewertet.



**Abbildung 3.7:** Exemplarische Einzelauswertung für ein Produkt mittels AttrakDiff [83]

Das aus dieser Bewertung berechnete Ergebnis ist in Abbildung 3.7 zu sehen und stellt den subjektiven Eindruck der Nutzer dar. Das orange gefärbte Rechteck repräsentiert die Varianz um den Mittelwert der Bewertungen sowohl in der hedonischen als auch in der pragmatischen Qualität.

Die beurteilten Dimensionen des Produkts sind laut Entwickler:

- **Pragmatische Qualität (PQ):** Sie beschreibt die Benutzbarkeit eines Produkts und verdeutlicht, wie gut der Nutzer seine Ziele mithilfe des Produkts erreichen kann.
- **Hedonische Qualität - Stimulation (HQ-S):** Menschen haben das Bedürfnis, sich weiterzuentwickeln. Diese Dimension bildet ab, inwieweit ein Produkt diese Entwicklung unterstützen kann, indem es neuartige, interessante und anregende Funktionalitäten, Inhalte, Interaktions- und Präsentationsstile bietet.
- **Hedonische Qualität - Identität (HQ-I):** Sie beschreibt, inwieweit ein Produkt seinem Nutzer ermöglicht, sich mit dem Produkt zu identifizieren.
- **Attraktivität (ATT):** Sie beschreibt eine globale Bewertung des Produkts auf der Basis der wahrgenommenen Qualität.

Die pragmatische und hedonische Qualität gehen zu fast gleichen Teilen in die Bewertung der Attraktivität ein. Mehr zur Motivation hinter der Entwicklung dieses standardisierten Fragebogens und der verwendeten Methodik findet sich in [84].

### 3.3.2 System Usability Scale (SUS)

Der SUS-Fragebogen ist ebenfalls eine standardisierte Methode, um subjektive Eindrücke der Nutzer zu unterscheiden. Er stammt aus dem Jahr 1986 und wurde ursprünglich entwickelt, um Terminalanwendungen zu bewerten. Er hat sich aufgrund seiner Einfachheit jedoch durchaus zu einem Standard für Evaluierungen von Schnittstellen entwickelt, auch wenn er oft als *quick and dirty* bezeichnet wird. Dieser Fragebogen besteht aus zehn Fragen, die mittels eines semantischen Differenzials in fünf Stufen (1 *stimme nicht zu* bis 5 *stimme voll zu*) beantwortet werden. Zwei der Fragen zielen auf die Erlernbarkeit ab und die restlichen acht auf die Benutzbarkeit des Systems [85]. Die Fragen sind abwechselnd positiv (ungerade Fragennummern) und negativ (gerade Fragennummern) formuliert. Bei der Wertung werden die ungeraden Fragen mit dem Wert der Skala abzüglich 1 gewertet und die geraden Fragennummern mit 5 abzüglich dem Wert der Skala. Dadurch sind jeweils höchstens  $5 \cdot 4 = 20$  Punkte für den positiven und negativen Teil der Fragen möglich. Die Gesamtpunktzahl wird mit dem Faktor 2,5 multipliziert. Somit liegt die maximal erreichbare Punktzahl bei 100 (40 Punkte mal 2,5). In [86] wurden 2324 SUS-Nutzerstudien empirisch untersucht und dabei ein Durchschnittswert von 70,1 ermittelt, mit einem Konfidenzintervall für  $\alpha = 0,01$  von 68,7 bis 71,5. Nimmt man diesen Mittelwert der Studie als Orientierungshilfe, so kann man ab einer darüberliegenden Punktzahl von einer überdurchschnittlich guten Benutzbarkeit des Systems sprechen.

## 3.4 Verwendete Software

In diesem Abschnitt wird eine Übersicht über verwendete Software geliefert. Hierbei werden zunächst die Softwarezwischenschichten eingeführt, die zum Austausch der Daten zwischen den entwickelten Komponenten zum Einsatz kamen. Des Weiteren werden die Softwarepakete zum Auslesen des Microsoft Kinect-Sensors sowie zum Verarbeiten der Punktwolkendaten kurz erläutert.

### 3.4.1 Internet Communication Engine (ICE)

Die ICE [87] bietet verschiedene Kommunikationsmodelle für die verteilte Programm-entwicklung. Die Schnittstellendefinition erfolgt in einer eigenen Metasprache, die mittels eines mitgelieferten Übersetzers in verschiedene Programmiersprachen (C++, .NET, Java, Python) und für unterschiedliche Plattformen (Linux, Windows, Android, iOS) direkt als Programmcodevorlage portiert werden kann. Diese Software spielt ihre

Entwicklungsvorzüge insbesondere bei über das Netz verteilten Anwendungen (wie Skype) - daher auch der Name Internet Communication Engine - aus.

Neben dem bereits sehr bekannten Client-Server-Kommunikationsprinzip bietet diese Software zum Beispiel das Publisher- und Subscriber-Konzept. Hierbei werden Nachrichten zu einem Thema vom Publisher veröffentlicht. Andere Programme können sich als Subscriber bei einem weiteren Programm, dem TopicManager, registrieren und erhalten dann zu diesem Thema veröffentlichte Informationen. Ein wesentlicher Vorteil dieses Konzepts ist die Modularität. Die Programme benötigen nur die Schnittstelleninformation und können dann vollkommen unabhängig voneinander entwickelt werden. Es können auch mehrere Subscriber ein Thema abonnieren und verarbeiten und diese Information selbst unter einem neuen Thema veröffentlichen.

Die Software unterstützt unter anderem synchrone und asynchrone verteilte Methodenaufrufe (engl.: Remote Procedure Call (RPC)), vergleichbar zum Simple Object Access Protocol (SOAP). Außerdem wird die Serialisierung und Deserialisierung der objektorientierten Daten für die Übertragung, den automatischen Verbindungsaufbau bei getrennter Verbindung, die parallele Verarbeitung, die Lastverteilung, die sichere Verbindung sowie die Protokoll-Kompressionen vereinfacht.

### 3.4.2 Realzeit-Datenbasis

Die Realzeit-Datenbasis (engl.: Real-Time Database (RTDB)) wurde von Matthias Göbel [88] für die Speicherung und Auswertung von Sensordaten entwickelt. Diese Software entstand im Rahmen der Steuerung von kognitiven Automobilen und wird deshalb auch mit KogMoRTDB abgekürzt. Die RTDB ist zusammen mit der Dokumentation sowie einigen Anwendungshinweisen unter [89] verfügbar. Die Software bietet die Möglichkeit, Echtzeitaufgaben und weniger zeitkritische Aufgaben miteinander zu synchronisieren. Die RTDB ist dabei nicht im Sinne einer SQL-basierten Datenbank zu verstehen und wird deshalb auch als Datenbasis bezeichnet. Das Grundprinzip gleicht eher einem geteilten Speicherbereich für Programme, die Daten schreiben und lesen können. Dabei werden die gewünschten Datenstrukturen in einem Ringpuffer gespeichert, dessen Größe von der Datenrate des schreibenden Programms abhängt und vom Anwender zu spezifizieren ist. Durch diesen Ringpuffer wird folglich eine Art Kurzzeitgedächtnis realisiert. Dadurch steht ein kurzer zeitlicher Verlauf der Informationen bereit, der durch die Größe des Ringpuffers bestimmt wird. In diesem Zusammenhang ist auch eine Aktualisierungsrate anzugeben, die meist mit der Datenrate übereinstimmt und die angibt, in welchen zeitlichen Abständen eine Aktualisierung des Datenschreibers zu erwarten ist. Zum einen kann diese Information für eine optimale

Ablaufstrategie (beispielsweise Earliest Deadline First (EDF)) auf einem Realzeitbetriebssystem genutzt werden und zum anderen kann damit überwacht werden, ob Informationen eines Datenschreibers noch aktuell sind beziehungsweise erkannt werden, ob eine schreibende Komponente ausgefallen ist. Andere Komponenten, die die Sensorinformationen zu höherwertigen Informationen weiterverarbeiten möchten, können die vorhandenen Daten aus dem Ringpuffer lesen und darauf basierend ihre Berechnungen ausführen und wiederum selbst Ergebnisse in die RTDB zurückschreiben.

#### **3.4.3 Microsoft Kinect Software Development Kit (SDK)**

Das Microsoft Kinect SDK ist die offizielle Software zum Auslesen der Sensordaten und zur Entwicklung von Programmen für den Kinect-Sensor. Dabei werden neben Hinweisen zur Entwicklung von Benutzerschnittstellen auch Programmbeispiele für die verschiedenen Funktionen mitgeliefert. Die Software kann neben den rohen Farb-, Infrarot-, Audio- und Tiefendaten auch Informationen zu detektierten Personen und deren ermittelten Posen in Form von Gelenkwinkeln und Positionen eines Skeletts liefern. Des Weiteren ist seit der Version 1.5 auch die Erkennung und Verfolgung von Gesichtsmerkmalen mittels eines dreidimensionalen Gesichtsmodells integriert.

#### **3.4.4 Punktwolken-Bibliothek**

Für die Verarbeitung der Daten des Kinect-Sensors wurde die Punktwolken-Bibliothek (engl.: Point Cloud Library (PCL)) verwendet, die unter [90] verfügbar ist und deren Entwicklung wesentlich von Radu Bogdan Rusu vorangetrieben wurde [91]. Diese Bibliothek bietet verschiedene Funktionen und Algorithmen, um eine Punktwolke zu verarbeiten. Inzwischen steht eine sehr große Entwickler- und Nutzergemeinschaft hinter dieser Bibliothek und es finden sich immer mehr Funktionen und Anwendungen, die mit dieser Bibliothek realisiert werden können.

# Kapitel 4

## Architektur zur multimodalen Mensch-Roboter-Interaktion

Dieses Kapitel gibt einen Überblick über verwandte Arbeiten zur multimodale Mensch-Roboter-Interaktion sowie mögliche Ziele dieser Interaktionsform. Es werden realisierte Interaktionskanäle zwischen Mensch und Roboter für ein industrielles sowie ein soziales Umfeld vorgestellt. Abschließend wird die entwickelte Architektur zur multimodalen Interaktion beschrieben.

### 4.1 Motivation

Die Motivation für die Implementierung einer eigenen Architektur zur multimodalen Mensch-Roboter-Interaktion war die Kombination der folgenden Anforderungen:

- Realzeitfähigkeit
- Kurzzeitspeicher
- Dienstorientierte Architektur
- Wissensbasierte Ablaufsteuerung

Die Realzeitfähigkeit ist dabei für eine flüssige Interaktion mit dem Benutzer notwendig. Die Aufnahme der Daten, deren Verarbeitung durch die Algorithmen sowie die Ausgabe an den Benutzer sollten idealerweise innerhalb eines Zeitfensters von bis zu 200 ms [92] abgeschlossen sein. Ansonsten sollte das System ein Feedback ausgeben, dass die Eingabe zwar erkannt wurde, jedoch noch verarbeitet werden muss.

Ein typisches Beispiel hierfür ist die bekannte und beliebte Sanduhr. Die Speicherung der Sensordaten in einem Kurzzeitspeicher ist nützlich für die Klassifikation von Eingaben mittels gleitender Fenster (engl.: sliding Windows). Dadurch müssen die Daten nicht vorher segmentiert werden. Allerdings können dadurch mehrfache Erkennungen einer Benutzeraktion auftreten. Diese Mehrfacherkennungen können durch eine geeignete Totzeit (circa 2-3 Sekunden) nach der ersten Erkennung unterdrückt werden (engl. multiple Supression). Die dienstorientierte Architektur ermöglicht eine einfache Einbindung sowie Austauschbarkeit von Netzwerkknoten, die dedizierte Aufgaben durch den angebotenen Dienst erledigen. Somit können besonders rechenintensive Algorithmen ausgelagert und lediglich die Rohdaten sowie das Ergebnis kommuniziert werden. Die dienstorientierte Verarbeitung pausiert die Prozesse automatisch, falls ein Modul ausfällt und setzt die Verarbeitung fort, wenn die benötigten Daten durch den Dienst des Moduls wieder zur Verfügung gestellt werden. Die wissensbasierte Ablaufsteuerung ermöglicht das Hinterlegen von Expertenwissen zum Ablauf von typischen Interaktionssequenzen sowie die Anbindung an Wissensdatenbanken und Ontologien zur Inferenz der nächsten Schritte.

Grundsätzlich ist für viele Sensormodule und Aktoren noch eine individuelle Anbindung an die Steuerungssoftware notwendig. Dies liegt an fehlenden einheitlichen Standards in der Robotik, die es laut [93] auch noch einige Zeit nicht geben wird. Solche Standards würden große Vorteile für die Entwicklung bringen, da viele Algorithmen dann auf diese standardisierten Schnittstellen zurückgreifen könnten und nicht speziell für die jeweilige zugrunde liegende Hardware angepasst werden müssten. Hierzu gibt es sehr viele Projekte in der Forschung und der Industrie, um Plattformen und Standards zu etablieren, um eine Abstraktion von der unterschiedlichen Hardware der verschiedene Roboter und Sensoren zu ermöglichen. Im nachfolgenden Abschnitt wird ein kurzer Überblick über verwandte Arbeiten bezüglich solcher Bemühungen für Robotersoftwareplattformen sowie multimodaler Verarbeitungsarchitekturen gegeben.

## 4.2 Verwandte Arbeiten

Im Bereich der Robotersoftwareplattformen gibt es immer wieder neue Entwicklungen, um besondere Anforderungen der Aufgabe oder spezielle Eigenschaften des Roboters zu unterstützen, die in bisher bestehenden Arbeiten noch nicht oder nicht wie gewünscht umgesetzt wurden. [94] trägt als Anspielung auf solche üblichen Eigenentwicklungen für Roboterplattformen sogar selbst den Namen Yet Another Robot Plattform (YARP). Die Plattform wurde für humanoide Roboter entwickelt und unterstützt

unterschiedliche Betriebssysteme wie Windows, Linux und QNX RTOS [95], da Gerätetreiber häufig immer noch nur für eine spezielle Plattform zur Verfügung stehen. So können die besten Merkmale der Betriebssysteme kombiniert werden. Neben der Modularität der Softwarekomponenten wird auch eine rechnerübergreifende Kommunikation unterstützt, um mehr Rechenleistung für das Gesamtsystem zur Verfügung zu haben. Dabei werden sogenannte Port-Objekte als Schnittstellendefinition verwendet. Zugrunde liegende Kommunikationsverfahren, die von YARP unterstützt werden, sind die Netzwerkprotokolle Transmission Control Protocol (TCP) und User Datagram Protocol (UDP). Des Weiteren können Nachrichten mittels Multicast an mehrere Empfänger geschickt werden und für die lokale Prozesskommunikation wird ein gemeinsam genutzter Speicher unterstützt. Seine Anwendung findet YARP beispielsweise bei den Robotern Kismet [96] und iCub [97].

Microsoft Robotics Developer Studio [98] ist ebenfalls ein Framework für die Entwicklung von Software für Roboter. Die Version 4 bietet dabei Unterstützung für asynchrone und zustandsgesteuerte Applikationen und eine dienstorientierte Laufzeit. Neben Tutorials stehen dem Anwender bereits entwickelte Komponenten zur Verfügung. Dazu gehört auch eine Simulationsumgebung mit Physiksimulation und einigen Robotern aus dem industriellen Umfeld, wie beispielsweise dem KUKA Leichtbauroboter [99], aber auch Roboterplattformen wie Mindstorms der Firma LEGO [21], der primär für Didaktik verwendet wird. Zusätzlich gibt es auch eine visuelle Programmiersprache, mit der sich vorhandene und selbstentwickelte Blöcke miteinander verschalten lassen, um die Ablaufsteuerung festzulegen. Seit der Version 4 ist auch der Microsoft Kinect als simulierter Sensor für die Entwicklung verfügbar.

Robot Operating System (ROS) [100] wird von Willow Garage gepflegt und mit der Unterstützung durch weitere Personen und Institute weiterentwickelt und ist ein weiteres sehr wichtiges Beispiel für Roboterplattformen. Ziel der Software ist es, ein Betriebssystem für Roboter zu entwickeln. ROS erleichtert die Entwicklung von Anwendungen durch Abstraktion von der eigentlichen Hardware, durch Visualisierungssoftware, Nachrichten- und Paketverwaltung, Treiber für Sensoren und weitere von der Entwicklungsgemeinschaft bereitgestellte Funktionen.

Das System wird bereits mit Robotern wie dem PR2 [101] von Willow Garage, dem Fraunhofer IPA Care-O-bot [15], dem Mindstorms Roboter von LEGO [21], NAO von Aldebaran Robotics [20] sowie dem Staubsaugerroboter der Firma iRobot [23] und weiteren Robotern verwendet. Hier zeigt sich schon das große Potenzial und der Wunsch der Forschungsgemeinschaft nach einem einheitlichen Roboterbetriebssystem. Obwohl das Haupteinsatzgebiet von ROS bisher vorwiegend Forschungs- und

Didaktikplattformen sind und es noch recht wenig bei den klassischen industriellen Roboterherstellern verbreitet ist, gibt es auch dazu bereits erste Bestrebungen [102], die Situation zu ändern. Hierzu sollen einige Komponenten von ROS auf ein industrielles Level bezüglich Robustheit und Performanz gebracht werden. Dabei soll das Beste aus der Welt der industriellen Roboteransteuerung und der höheren Verarbeitungs- und Kontrollebene von ROS kombiniert werden.

Ein aktueller Überblick über Software zum vereinfachten Datenaustausch zwischen Komponenten wird auch in [103] und [104] präsentiert. Dabei kommen meist weitere Softwarezwischen-schichten (engl.: middlewares) zur Anwendung, die dazu dienen, die Kommunikation zwischen den Komponenten zu vereinfachen. Insbesondere für die Netzwerkkommunikation werden CORBA [105] oder ICE [87] eingesetzt. In der Automobilbranche wird hierfür zunehmend Etch [106] verwendet. Dabei wird dem Anwender insbesondere die Übermittlung, Serialisierung und Deserialisierung von Objekten sowie der notwendige Verbindungsaufbau abgenommen.

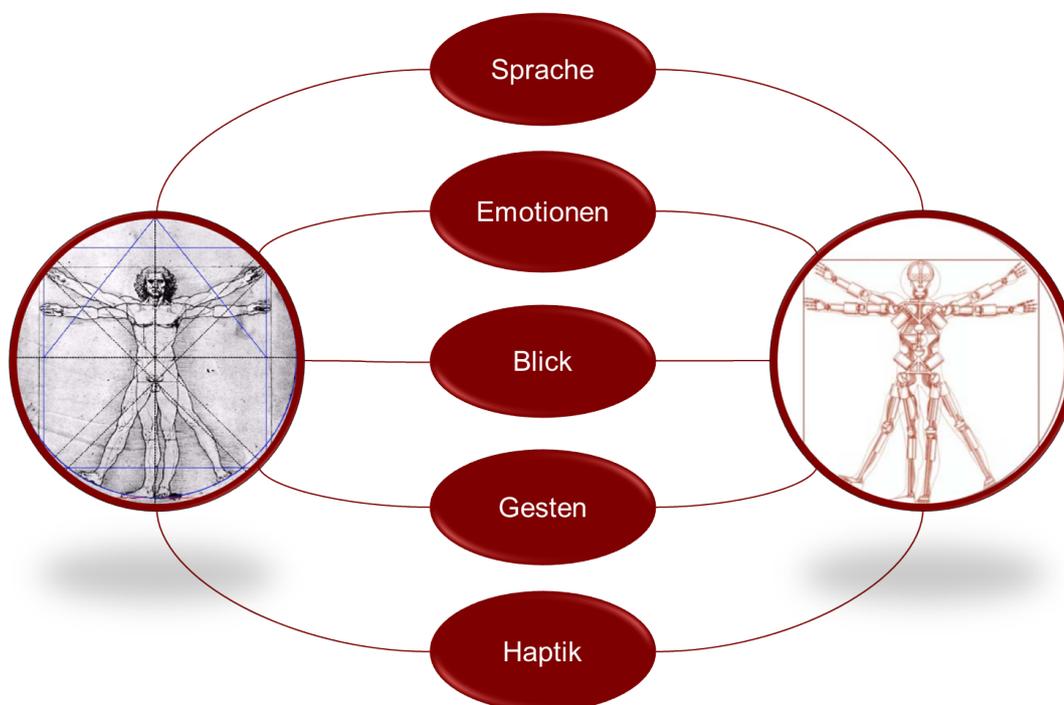
Neben diesen speziell für Roboter entwickelten Softwareplattformen gibt es auch Softwarestrukturen, deren primärer Verwendungszweck die Verarbeitung von multimodalen Daten ist. Ein Überblick vorhandener Architekturen wird in [107] präsentiert und anhand selektierter Merkmale verglichen. Die Architekturen werden dabei bezüglich des Zusammenspiels der Komponenten dahingehend unterschieden, ob diese durch eine Zustandsmaschine oder intelligente zusammenarbeitende Agenten orchestriert werden. Außerdem wird die Wiederverwendbarkeit der Architektur dahingehend betrachtet, ob ein Entwicklungswerkzeug verfügbar ist oder die Architektur zumindest eine Schnittstelle zur Programmierung (Application Programming Interface (API)) hat oder sogar visuell programmiert werden kann. Zusätzlich wird betrachtet, ob der Code quelloffen ist und sich leicht erweitern oder einbinden lässt.

### 4.3 Multimodale Interaktion

#### 4.3.1 Definition

Die multimodale Interaktion ist ein sehr weites Forschungsgebiet. Das Gebiet reicht von der kognitionspsychologischen Grundlagenforschung, die sich mit der Aufnahme und der Verarbeitung multimodaler Eingaben beim Menschen beschäftigt, bis hin zur Entwicklung von Algorithmen und Architekturen zur technischen Umsetzung.

Im Allgemeinen versteht man unter Multimodalität die Möglichkeit, verschiedene Modalitäten für die Interaktion zu nutzen. Im Zusammenhang mit dem Menschen wird



**Abbildung 4.1:** Modalitäten zwischen Mensch und Maschine

dieser Begriff häufig auf die Sinne (Hören, Sehen, Riechen, Schmecken, Tasten) bezogen, weshalb auch von Sinnesmodalitäten gesprochen wird. Dieses Verständnis von Multimodalität kann auch auf technische Systeme übertragen werden. Dadurch soll eine multimodale Interaktion und somit eine natürlichere Art der Interaktion zwischen Mensch und Maschine geschaffen werden, wie man sie von der Interaktion zwischen Menschen kennt. Da in dieser Arbeit die Interaktion mit Robotern im Fokus steht, wurden jedoch der Geruchs- und der Geschmackssinn nicht weiter betrachtet. Die im Rahmen der Projekte dieser Arbeit untersuchten Modalitäten zwischen Mensch und Maschine sind in Abbildung 4.1 zu sehen.

Eine Möglichkeit, die Modalitäten zu klassifizieren, ist die Richtung des Informationsflusses und somit Eingabe- und Ausgabemodalitäten beziehungsweise wahrnehmende und agierende Modalitäten zu unterscheiden [108]. Eine der sehr frühen Arbeiten zu diesem Thema aus dem Jahr 1980 nutzte bereits Sprache und Gesten als Eingabemodalitäten [109]. Eine weitere Möglichkeit zur Klassifikation besteht darin, zu unterscheiden, ob die Kanäle gleichzeitig – also parallel – Informationen übertragen. Ein typisches Beispiel ist hier die Bestellung von Getränken in einer lauten Umgebung. Dabei wird zusätzlich zu der gesprochenen Anzahl diese auch noch per stati-

## 4. ARCHITEKTUR ZUR MULTIMODALEN MENSCH-ROBOTER-INTERAKTION

---

scher Handgeste mitcodiert und somit die Robustheit durch redundante Informationen gesteigert. Alternativ können verschiedene Modalitäten genutzt werden, um die Informationseingabe für den Benutzer komfortabler zu gestalten. Hierbei wählt der Nutzer die Modalität zur Informationsübertragung, die ihm für den aktuellen Kontext optimal erscheint.

Laut [107] kann durch multimodale Systeme eine gesteigerte Robustheit erreicht werden. Hierfür müssen die unterschiedlichen Informationskanäle kombiniert werden. Des Weiteren kann mit der Multimodalität eine bessere Bedienbarkeit des Systems erreicht werden, wenn Informationen über den aktuellen Benutzer und den Kontext berücksichtigt werden.

Obwohl man vermutlich bei multimodalen Benutzerschnittstellen zunächst an eine höhere Effizienz bei der Bewältigung von Aufgaben im Vergleich zu unimodalen Schnittstellen denkt, scheinen die Aufgaben, zumindest laut [110] und dessen Studien, nur um einen Faktor von circa 10 % schneller erledigt zu werden. Benutzer mit multimodaler Interaktionsmöglichkeit machten zudem 36 % weniger Fehler und laut den Studien bevorzugen über 95 % die multimodale Interaktion gegenüber der unimodalen, da der Nutzer hier selbst die bevorzugte Modalität wählen kann. Hierbei wird eine multimodale Benutzerschnittstelle so definiert, dass zwei oder mehr Eingaben (beispielsweise Sprache, Gesten, Blick, Kopf- und Körperbewegungen) zusammen verarbeitet werden und eine multimediale Ausgabe generieren.

### 4.3.2 Ziele der multimedialen Interaktion

Wie es sich aus der obigen Definition bereits ergibt, können die Ziele, die mit der multimodalen Interaktion erreicht werden sollen, recht unterschiedlich sein. Zum einen kann die Benutzerschnittstelle für den Anwender komfortabler zu bedienen sein, indem er selbst die Eingabemodalität wählen kann. Ein Beispiel hierfür sind moderne Smartphones, bei denen eine Texteingabe über eine virtuelle Tastatur erfolgen kann, aber der Text auch diktiert werden kann und erst mittels Spracherkennung in Text umgewandelt wird. Falsch erkannte Textzeilen können dann wiederum mittels der virtuellen Tastatur korrigiert werden. Hierbei werden die Modalitäten sequenziell und nach Belieben des Nutzers verwendet.

Eine andere Möglichkeit ist die Bestätigung von Aktionen beispielsweise durch eine sprachliche Äußerung und gleichzeitiges Kopfnicken. Hierbei muss das System parallel den akustischen und den visuellen Kanal verarbeiten und die Ergebnisse zu einer Antwort fusionieren. Hierbei ist das Hauptziel der multimodalen Interaktion eher

die Robustheit, da beispielsweise eine nicht erkannte Spracheingabe durch die Kopfgebe kompensiert werden kann.

Obwohl durch multimodale Interaktion auch eine effizientere Abwicklung von Aufgaben möglich sein kann, lassen die Ergebnisse in [110] dies eher als positiven Nebeneffekt erscheinen. Hierbei kommt es darauf an, die optimale Modalität für die Aufgabe zu verwenden beziehungsweise das System muss aus den multimodalen Eingaben die relevanten Informationen extrahieren, um dem Benutzer weitere Eingaben, Wiederholungen und damit letztendlich auch Zeit zu ersparen.

Aus diesem Grund liegt der Fokus der multimodalen Interaktion in dieser Arbeit auf dem Anbieten verschiedener Modalitäten, aus denen der Benutzer die jeweils geeignete wählen kann, wodurch er einen höheren Bedienkomfort hat. Werden die Modalitäten parallel verwendet, so ist eine Fusion notwendig. Dies kann entweder früh in der Verarbeitungskette auf Ebene der Merkmale geschehen oder später anhand von semantischer Information. Die Fusion der Kanäle wird in dieser Arbeit nicht weiter betrachtet, da von einer sequenziellen Verwendung der Modalitäten ausgegangen wird.

Neben den oben genannten Zielen einer multimodalen Interaktion hat der Anwendungshintergrund einen großen Einfluss auf die Wahl der richtigen beziehungsweise vom Anwender erwünschten Modalitäten. Im Fall dieser Arbeit wurden ein soziales Umfeld sowie ein industrielles Umfeld betrachtet. Im sozialen Umfeld steht der Spaß und der Unterhaltungsfaktor bei der Interaktion im Vordergrund. Dagegen sind im industriellen Umfeld hauptsächlich die Produktivität sowie die Funktionalität relevant. Deshalb sind Emotionen und Gesten eher im sozialen Umfeld zu berücksichtigen. Für das soziale Umfeld wurde ein Szenario für die Bestellung von Getränken sowie ein Spiel zur Untersuchung der Interaktion selektiert. Diese werden in Abschnitt 7.2 detaillierter beschrieben. Für das industrielle Umfeld wurde eine hybride Fertigungsstation in einer kognitiven Fabrik ausgewählt sowie eine robotergestützte Verpackstation, die in Abschnitt 7.3 weiter erläutert werden.

Im Folgenden werden Kommunikationskanäle für die multimodale Architektur erläutert, die im Rahmen der Zusammenarbeit innerhalb des interdisziplinären Forschungsverbunds CoTeSys entstanden sind.

### 4.3.3 Sprache

Sprache ist ein in der zwischenmenschlichen Kommunikation häufig genutzter Kanal, um eine große Menge an Informationen zu transportieren. Für die Erkennung der Sprache werden meist HMM verwendet. Dabei wird die Sprache in Wörter zerlegt

#### 4. ARCHITEKTUR ZUR MULTIMODALEN MENSCH-ROBOTER-INTERAKTION

---

und die wahrscheinlichste Phonemfolge der HMMs ergibt dann das vermutlich gesprochene Wort. Es gibt zwei unterschiedliche Ansätze für die Spracherkennung. Zum einen die Erkennung eines beschränkten Wortschatzes. Hierfür wird eine Grammatik in Backus-Naur-Form (BNF) [111] aufgestellt und dadurch erlaubte Kommandosequenzen definiert. Dieser Ansatz reduziert die Möglichkeiten an zu erkennendem Vokabular und ist daher meist robuster in der Anwendung. Er wird beispielsweise bei der Spracherkennung mittels Microsoft Kinect SDK verwendet. Bei Kommando-basierter Spracherkennung ist es hilfreich, wenn der Nutzer die möglichen Kommandos visualisiert bekommt oder zumindest per Hilfe abrufen kann. Die andere Möglichkeit basiert auf der Erkennung von natürlicher Sprache. Dieser Ansatz ermöglicht prinzipiell die Erkennung beliebiger Spracheingaben. Dadurch ist diese Variante für den Benutzer am komfortabelsten, da er sich eben keine Kommandosequenzen merken muss. Aufgrund des größeren Wortschatzes muss für eine bessere Erkennungsleistung häufig ein allgemeines Sprechermodell an den jeweiligen Sprecher adaptiert werden. Dies geschieht üblicherweise durch Vorlesen von bekannten Testsätzen. Eine Anwendung dieser natürlichen Spracherkennung ist die Software Dragon Naturally Speaking [112]. Innerhalb dieser Arbeit wurden sowohl Erkennen basierend auf Grammatiken mittels DialogOS [113] und der Microsoft Speech Recognition API [114] als auch natürlichsprachliche Erkennen mittels Speech Interpretation and Recognition Interface (Siri) von Apple beziehungsweise der Implementierung der WebSpeech-API [115] von Google verwendet. Die Aktivierung der Spracherkennung erfolgt im industriellen Umfeld durch einen Tastendruck und im sozialen Umfeld durch Aufmerksamkeitsdetektion (Anblicken des Roboterkopfs) oder automatisch nach einer gestellten Frage, da hier eine Antwort des Benutzers erwartet wird. Beendet wird die Spracherkennung entweder nach dem Überschreiten einer zuvor definierten Zeitdauer oder nach einer längeren Sprachpause. Neben der Erkennung der Sprache müssen aus dem erkannten Text die Funktion sowie deren Parameter extrahiert werden. Bei Kommando-basierter Sprache ist dies recht einfach zu realisieren, da hier einfach die Grammatik entsprechend konstruiert werden kann. Die Verarbeitung von natürlichsprachlichen Eingaben bildet ein eigenes Forschungsgebiet, das sich lediglich mit dem Verstehen der natürlichen Sprache auseinandersetzt. Doch auch hier gibt es inzwischen Werkzeuge, die die Arbeit erleichtern, wie beispielsweise [116]. Erwähnenswert ist hier auch das Projekt WIT [117], das eine Webschnittstelle bietet, um anhand von bereitgestellten und annotierten Beispielsätzen die Intention des Benutzers zu extrahieren und somit natürliche Sprache in strukturierte Daten umzuwandeln. Bei Siri wurde dies gelöst, indem zunächst mittels einer Schlüsselworterkennung die Funktion selektiert wird und im Anschluss

daran anhand von regulären Ausdrücken die zugehörigen Parameter für die Funktion extrahiert werden. Spracheingaben des Benutzers, die keines der Schlüsselwörter enthalten beziehungsweise keine gültige Parameterextraktion erlaubt haben, können zur späteren Analyse gespeichert werden und gegebenenfalls in das gültige Vokabular aufgenommen werden.

Für die sprachbasierte Informationsübertragung vom Roboter zum Menschen wurde Sprachsynthese verwendet. In dieser Arbeit wurde hierfür eine kommerzielle Sprachsynthese von IVONA [118] beziehungsweise DialogOS [113] verwendet. Dabei konnten sowohl unterschiedliche Sprachen (Englisch, Deutsch) als auch unterschiedliche Sprecher (männlich, weiblich) selektiert werden.

#### 4.3.4 Emotionen

Durch die Zusammenarbeit mit Stefan Sosnowski war eine Anbindung an den Roboterkopf Emotion-Display with Dynamic Intuitive Expressions (EDDIE) [119] zur Darstellung von Emotionen möglich. Dieser Roboterkopf erlaubt die Darstellung der sechs Basisemotionen (Ärger, Freude, Überraschung, Traurigkeit, Angst und Ekel) [120] und macht die Interaktion besonders im sozialen Umfeld unterhaltsamer für den Benutzer [121].

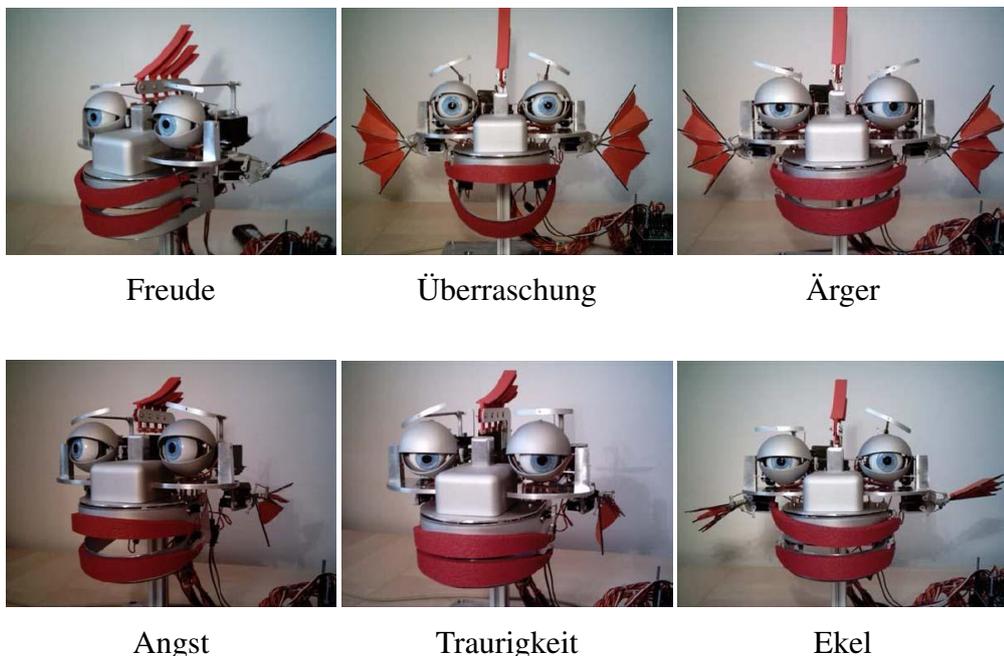
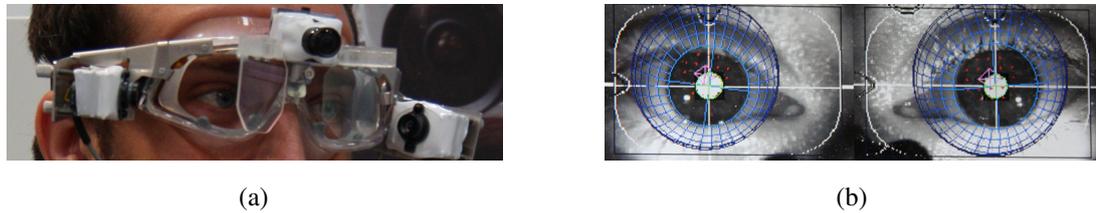


Abbildung 4.2: Roboterkopf EDDIE [119] zeigt sechs Basisemotionen [122].

Diese Darstellungsmöglichkeit der Emotionen wurde bei Anwendungen zur Interaktion im sozialen Umfeld (siehe Abschnitt 7.2) eingebunden.

### 4.3.5 Blick

Die Blickdatenerfassung wird häufig in Nutzerstudien verwendet, wobei Karten von Bereichen erstellt werden, die von besonderem Interesse sind. Dadurch kann visualisiert werden, wo der Nutzer beispielsweise auf einer Webseite besonders häufig hinsieht und sich somit Werbung gut platzieren lässt. Den Blick des Nutzers zu erfassen und diese Daten für die Interaktion zu verwenden, ist hingegen noch ein recht junges Forschungsthema in der Mensch-Maschine-Interaktion. Die Anwendungsmöglichkeiten und Forschungsgebiete sind dabei recht vielfältig. Eine Anwendung ist beispielsweise die freihändige Interaktion mit Anzeigen im Fahrzeug. Dieser Anwendungsfall wird in [123] näher beschrieben. Außerdem kann der Blick als alternative Eingabemöglichkeit für Zeichen verwendet werden, wie in [124], um damit beispielsweise Menschen mit Behinderung eine alternative Möglichkeit zur Zeicheneingabe bereitzustellen. Benutzerschnittstellen, die die Blickdaten auswerten, um die Aufmerksamkeit des Nutzers zu interpretieren, stellen eine weitere Anwendungsmöglichkeit dar. Es ist hier beispielsweise denkbar, beim Blick auf bestimmte Bereiche einer GUI diese für eine bessere Lesbarkeit durch Vergrößerung hervorzuheben. Eine weitere Möglichkeit ist, bei mehreren Monitoren den Mauscursor auf den aktiv betrachteten Bildschirm zu setzen beziehungsweise das Fenster, das gerade betrachtet wird, zu fokussieren oder wie bei neueren Mobiltelefonen den Bildschirm nur bei Betrachtung zu beleuchten und damit Strom zu sparen. Auch wenn der Blick als Eingabemedium grundsätzlich viele Vorteile bietet, wie beispielsweise die sehr schnelle, intuitive und freihändige Auswahl von Objekten, so muss berücksichtigt werden, dass der Mensch den Blick auch dafür nutzt, seine Umgebung zu erkunden. Daher sollte ein ideales System, das den Blick als Eingabekanal verwendet, verhindern, dass Aktionen unbeabsichtigt ausgelöst werden, wenn der Nutzer sich lediglich umschaute. Eine Möglichkeit, dies zu vermeiden, ist, die Verweildauer des Blicks bei der Interpretation der Interaktion zu verwenden. Die Funktion hinter der betrachteten Oberfläche wird dann erst aktiviert, wenn der Nutzer diese länger ansieht. Diese Variante wurde in dieser Arbeit zur Interaktion verwendet. Eine andere Möglichkeit ist, den Blick nur zur Selektion der Funktion zu verwenden, während die eigentliche Aktivierung über eine weitere Modalität, wie beispielsweise einen Knopfdruck, erfolgt.



**Abbildung 4.3: EyeSeeCam [125]** - (a) Kopfgebundenes Blickerfassungsmodul (b) Pupillenverfolgung durch die EyeSeeCam-Software

Des Weiteren gibt es unterschiedliche Techniken und Systeme, um den Blick eines Nutzers zu erfassen. So wurden bereits 1963 mithilfe von Spulen auf einer Kontaktlinse und einem magnetischen Feld sehr präzise Augenbewegungen gemessen [126]. Methoden, die auf der Videookulographie (VOG) beruhen, extrahieren die Blickdaten aus Videobildern und sind somit nicht so invasiv wie das vorherige Verfahren. Dabei kann man zwischen kopfgebundenen Systemen (Helme oder Brillen), Systemen, in denen der Kopf eingespannt wird und Systemen, die aus einiger Entfernung den Blick erfassen können, wie beispielsweise Kameras an einem Monitor [127], unterscheiden.

Für die Blickdatenerfassung wurde in dieser Arbeit mit der Unterstützung durch Stefan Kohlbecher und Stanislav Bardins die EyeSeeCam [125] als kopfgebundenes System verwendet, die ursprünglich in der medizinischen Diagnostik von Augenkrankheiten eingesetzt wird. Wie in Abbildung 4.3 (a) zu sehen ist, hat das verwendete Modell drei Kameras. Hierbei werden zwei Kameras verwendet, um über die Spiegelungen die Pupillenzentren des linken und des rechten Auges zu detektieren (siehe Abbildung 4.3 (b)). Die dritte Kamera ist mittig als Kamera zur Aufzeichnung des Sichtfelds des Trägers montiert. In diesem aufgezeichneten Kamerabild kann dann beispielsweise der berechnete Blickpunkt eingezeichnet werden. Außerdem können durch die Kamera Infrarot-LEDs gefunden werden, die eine Ebene zur Interaktion kennzeichnen. Dadurch lässt sich der Schnittpunkt des errechneten Blickvektors mit der detektierten Ebene berechnen, um sensitive Interaktionsflächen zu aktivieren, die im folgenden Abschnitt als Interaktionsmedium beschrieben werden. Die auf die Ebene transformierten Blickdaten wurden mit 120 Hz in die Realzeitdatenbank geschrieben.

Der Blick des Roboters wurde ebenfalls als Kommunikationskanal für das soziale Anwendungsfeld betrachtet. In einer Gruppe kann der Roboter dadurch beispielsweise den nächsten Interaktionspartner selektieren und dies seinem Gegenüber kommunizieren. Außerdem kann der Roboter dem Benutzer damit mitteilen, ob das System momentan in der Lage ist, Anweisungen aufzunehmen, und somit seine interne

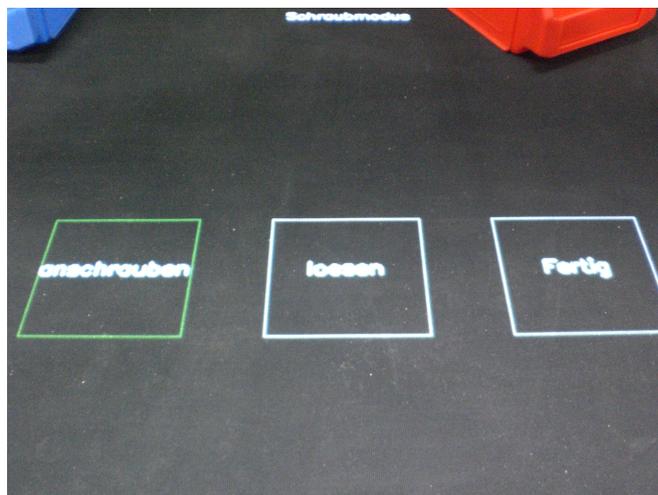
## 4. ARCHITEKTUR ZUR MULTIMODALEN MENSCH-ROBOTER-INTERAKTION

---

Auslastung und Aufmerksamkeit kommunizieren. Hierbei kann er Bereiche ansehen, die gerade von den Bildverarbeitungsprogrammen verarbeitet werden oder den Nutzer regelrecht anstarren, wenn er auf eine Eingabe von ihm wartet. In [128] wird eine Methode präsentiert zur automatischen Blickdatenauswertung. Dabei wurde der Blick von zwei Menschen, die sich kennen, in einem Gespräch aufgezeichnet. Im aufgezeichneten Szenenbild wurde mittels der faceAPI [129] automatisch das Gesicht des Gesprächspartners detektiert und der Blickpunkt mit dem Gesichtsmodell abgeglichen. Durch Analyse der Daten kann ermittelt werden, wie lang und wie häufig sich die Partner ins Gesicht blicken und welche Bereiche im Gesicht betrachtet wurden. Das im Rahmen dieser Arbeit entwickelte Modell zur Blicksteuerung von Robotern in Dialogsituationen findet sich in Kapitel 5.

### 4.3.6 Sensitive Interaktionsflächen

Neben den in der Industrie weit verbreiteten haptischen Eingabemethoden (Knöpfe, Tasten etc.) wurden im Rahmen des Projekts JAHIR und in Zusammenarbeit mit Alexander Bannat sogenannte sensitive Interaktionsflächen entwickelt. Hierbei wurden für den industriellen Einsatz konfigurierbare sensitive Eingabeflächen von oben herab auf eine Arbeitsfläche projiziert. Exemplarisch sind diese in Abbildung 4.4 zu sehen. Im



**Abbildung 4.4:** Drei auf die Arbeitsoberfläche projizierte sensitive Interaktionsflächen

Gegensatz zu echten Schaltern können diese nicht haptisch gedrückt werden. Durch das projektionsbasierte Verfahren sind diese Interaktionsflächen aber nahezu unverwundlich und daher auch besonders für den industriellen Einsatz geeignet. Die Be-

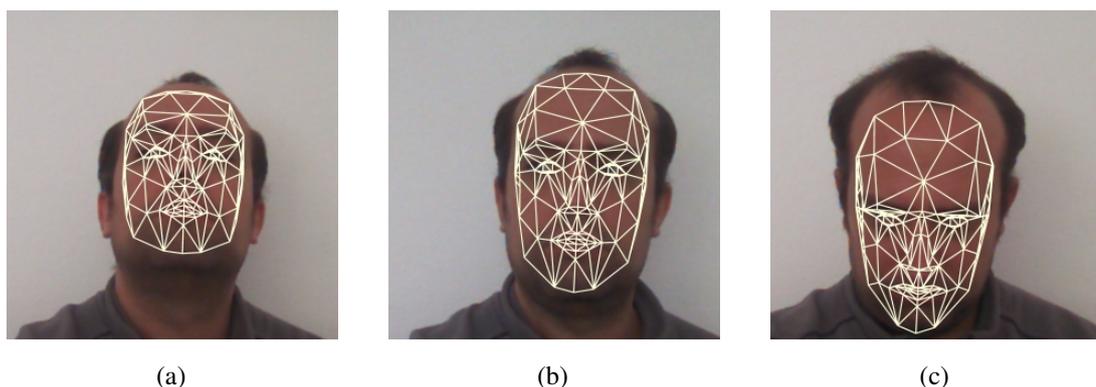
tätigung dieser Interaktionsflächen kann auf drei Wegen geschehen. Entweder durch Sprechen des projizierten Textes mittels Spracherkennung oder durch das Einbringen der Hand in die Sensorfläche in der Nähe der Arbeitsoberfläche. Dies kann entweder mittels eines ebenfalls von oben aufgenommenen Kamerabildes und anschließender Hautfarbendetektion [130] oder durch eine Tiefenbildkamera ausgewertet werden. Zusätzlich sollte auch hier eine Zeitschwelle für die Auslösung der Aktion verwendet werden. Damit kann eine nur über die Interaktionsfläche bewegte Hand von einer bewussten Aktivierung der Funktion unterschieden werden. Des Weiteren kann die Interaktionsfläche mittels blickbasierter Interaktion aktiviert werden.

### 4.3.7 Gesten

Gesten sind neben dem Blick ein sehr wichtiger Teil der nonverbalen Kommunikationsmittel [131] und werden deshalb in der zwischenmenschlichen Kommunikation häufig verwendet.

In Zusammenarbeit mit Christoph Mayer und Tobias Rehl wurden Kopfgesten zur Kommunikation mit den Robotern verwendet, um Befehle über fehleranfällige Übertragungskanäle, wie Spracheingabe, zu bestätigen. Dies stellt eine intuitive Möglichkeit zur Zustimmung oder Ablehnung in der zwischenmenschlichen Kommunikation dar.

Abbildung 4.5 zeigt Bilder aus der Kopfgestensequenz Nicken, mit eingezeichnetem 3D-Gesichtsmodell. Für die Registrierung des 3D-Gesichtsmodells wurde zuerst



**Abbildung 4.5: Bilder aus der Kopfgestensequenz Nicken -**

die Software von [132] und später das Microsoft Kinect SDK mit der integrierten Gesichtsverfolgung verwendet.

#### 4. ARCHITEKTUR ZUR MULTIMODALEN MENSCH-ROBOTER-INTERAKTION

---

Aus dem 3D-Gesichtsmodell wurden die folgenden fünf Merkmale als Grundlage für die Beobachtungen des HMMs verwendet. Die  $x$  und  $y$  Position des eingepassten 3D-Gesichtsmodells in der Bildebene sowie die drei Rotationswinkel  $\alpha$ ,  $\beta$ ,  $\gamma$  des Modells. Allerdings wurden diese Werte nicht direkt, sondern deren Änderungen über die Zeit als Beobachtungen verwendet. Damit ergibt sich also der Beobachtungsvektor  $\mathbf{O}_t$  zum Zeitpunkt  $t$  zu:

$$\mathbf{O}_t = \begin{pmatrix} \Delta x = x_t - x_{t-1} \\ \Delta y = y_t - y_{t-1} \\ \Delta \alpha = \alpha_t - \alpha_{t-1} \\ \Delta \beta = \beta_t - \beta_{t-1} \\ \Delta \gamma = \gamma_t - \gamma_{t-1} \end{pmatrix}$$

Für die Evaluierung des Systems wurden die Daten von 14 Personen aufgezeichnet, die jeweils drei verschiedene Kopfgesten (Nicken, Schütteln, Neutral) darboten. Für jede Geste und Person wurden zwei Sequenzen aufgezeichnet. Die Daten wurden in sechs disjunkte Mengen eingeteilt. Es wurden fünf Mengen zum Training und eine zum Evaluieren verwendet. Dies kann sechsmal wiederholt werden, wobei jede Menge einmal zum Evaluieren und nicht zum Training verwendet wird. Dabei spricht man von einer sechsfachen Kreuzvalidierung. Diese Methode wird häufig verwendet, wenn die Anzahl der Datensätze für eine feste Gruppierung in Trainings- und Evaluierungskorpora [133] zu gering ist. Dann wird der Mittelwert aus diesen sechs Evaluierungen bestimmt, wie in Tabelle 4.1 zu sehen:

Erkannt \ Label	Label		
	Nicken	Neutral	Schütteln
Nicken	100 %	0 %	0 %
Neutral	11 %	89 %	0 %
Schütteln	0 %	6 %	94 %

**Tabelle 4.1:** Mittelwerte der sechsfachen Kreuzvalidierung für die Kopfgestenerkennung mit kontinuierlichen HMMs [134].

Somit ergibt sich für das kontinuierliche HMM mit fünf Zuständen ein mittlerer Fehler von 5,67 % .

Für weitere Gesten zur Interaktion wurde das in [135] entwickelte Gestenerkennungssystem eingesetzt sowie Dynamic Time Warping aus [136] verwendet. Für die

Trainingsphase individueller Gesten wurde die Version so modifiziert, dass erst nach dem Überschreiten eines Bewegungsenergieschwellwerts die Daten für zwei Sekunden aufgezeichnet wurden.

### **4.4 Architektur zur multimodalen Interaktion**

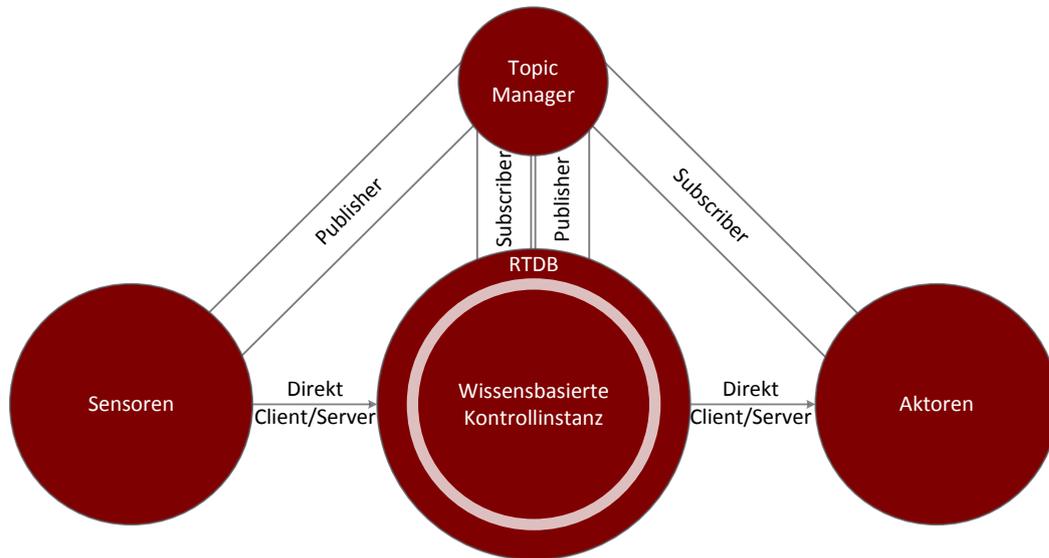
Die entwickelte Software-Architektur zur multimodalen Interaktion mit Robotern [137] basiert auf einer wissensbasierten Kontrollinstanz sowie der RTDB und ICE zur Anbindung der Sensoren und Aktoren. Die wissensbasierte Kontrollinstanz sowie die RTDB laufen auf einem leistungsstarken Rechner. Der im Rahmen dieser Arbeit verwendete Rechner lief mit dem Betriebssystem Ubuntu und verfügte über vier Kerne, die mit 2,2 GHz getaktet waren, sowie über vier Gigabyte Arbeitsspeicher. Sensoren mit großen Datenströmen werden idealerweise direkt an diesen Rechner angebunden, um die Datenströme nicht erst über das Netzwerk zu leiten. Die Anbindung der anderen Rechenknoten im Netzwerk erfolgt mittels ICE. Die Sensordaten werden mit Zeitstempeln versehen, wobei die Zeit im lokalen Netzwerk mittels Network Time Protocol (NTP) synchron gehalten wird. Dadurch liegt der Zeitunterschied laut [138] bei maximal 200 Mikrosekunden. Die Zeitstempel erlauben eine Synchronisierung der Daten in der RTDB zur weiteren Verarbeitung. Außerdem ist es durch das Ringpuffer-Konzept der RTDB möglich, auch Daten aus der Vergangenheit bei der Verarbeitung zu berücksichtigen, soweit diese noch im Puffer vorliegen. Die Architektur kann auch ohne Aktoren verwendet werden, wobei die Daten aufgezeichnet und für eine spätere Auswertung wiedergegeben werden können. Dies fand beispielsweise für Experimente zur zwischenmenschlichen Interaktion im Rahmen des Projekts Multimodal Dialog System (MuDiS) [139] Anwendung.

In Abbildung 4.6 ist das Kommunikationsprinzip zur Anbindung der Sensoren und der Aktoren über ICE und RTDB an die wissensbasierte Kontrollinstanz dargestellt.

Wie bereits erwähnt, wird für die Kommunikation zwischen den Komponenten eine Kombination aus der RTDB und ICE verwendet. Die Sensoren und Aktoren werden entweder direkt an den Rechner mit der RTDB und der Kontrollinstanz angebunden oder über ICE als Client oder Server beziehungsweise als Publisher oder Subscriber über einen TopicManager. Die Anbindung über den TopicManager ermöglicht eine Verbindung zwischen Sensor und Aktor für Regelprozesse. Die mit Zeitstempeln versehenen externen Sensordaten werden in definierten Datenstrukturen in die RTDB geschrieben. Weitere Verarbeitungskomponenten können auf die in der RTDB gepufferten Daten zugreifen und daraus semantisch höherwertige Informationen generieren.

#### 4. ARCHITEKTUR ZUR MULTIMODALEN MENSCH-ROBOTER-INTERAKTION

---



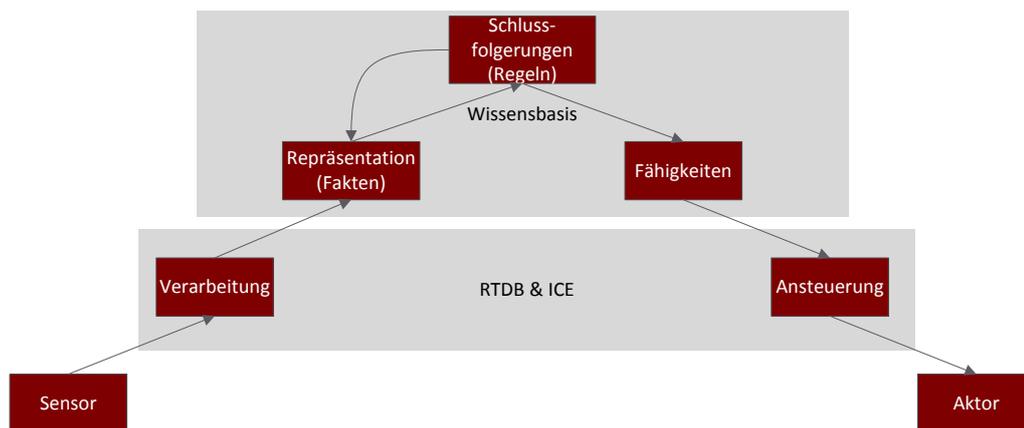
**Abbildung 4.6:** Prinzip der Kommunikation der Sensoren sowie Aktoren mit der wissensbasierten Kontrollinstanz über ICE und RTDB

Dies kann beispielsweise eine Bildvorverarbeitung sein, gefolgt von einer Gesichtsdetektion und einer anschließenden Gesichtsidentifikation. Die wissensbasierte Kontrollinstanz greift die gewünschten Daten (beispielsweise erkannte Person) aus der RTDB ab. Mittels des hinterlegten Expertenwissens werden dann die externen Aktoren benachrichtigt und angesteuert.

Für die wissensbasierte Kontrollinstanz wurde Java Expert System Shell (JESS) [140] sowie Prolog (franz.: **P**rogrammation en **l**ogic) [141] in der SWI-Prolog-Implementierung [142] eingesetzt. Diese beiden Logiksprachen sind Repräsentanten für Logiken erster Ordnung. Die Systeme arbeiten auf einer Wissensbasis, in der sowohl Fakten als auch Regeln hinterlegt sind. Die gewünschte Ablaufsteuerung wird durch Regeln repräsentiert. Die (Äquivalenz-) Regeln für den Ablauf werden von Experten erstellt, weshalb diese Ansätze allgemein auch als Expertensysteme bezeichnet werden. Eine in Prolog notierte Äquivalenzregel sieht beispielsweise folgendermaßen aus:

```
orderDialog(P,D) :-
isIdentified(P), isOrdered(D), isConfirmed(D).
```

Durch die Regel werden aus dem Prädikat auf der linken Seite mit den zwei Variablen  $P$  und  $D$  die drei Prädikate auf der rechten Seite erzeugt. Die Variablen stehen für Informationen, die erst durch die Sensoren beziehungsweise durch die Verarbeitungskomponenten mit expliziten Werten belegt werden. Im Beispiel wären dies eine identifizierte Person für  $P$  und das gewünschte Getränk  $D$ . Die Verarbeitungskette von den Sensoren über die Verarbeitung und die Repräsentation als Fakten für Schlussfolgerungen in der wissensbasierten Kontrollinstanz bis hin zur Ansteuerung der Aktoren ist in Abbildung 4.7 schematisch dargestellt. Dabei wird die Wissensbasis bei jeder Aktualisierung der Fakten auf anwendbare Regeln hin überprüft. Diese Regeln generieren entweder weitere Fakten oder münden im Aufruf von Befehlen der hinterlegten Fähigkeiten der Aktoren.



**Abbildung 4.7:** Ansteuerung der Aktoren aufgrund von Schlussfolgerungen in der Wissensbasis basierend auf den gelieferten Fakten der Sensoren

Dazu wurden Prolog und JESS an die RTDB angebunden, um bei der Entstehung semantischer Daten die zugehörigen Prädikate in die Wissensbasis einzufügen. Des Weiteren wurde auch der Rückkanal implementiert, um die durch Schlussfolgerung generierten Befehle für die Fähigkeiten der Aktoren wieder in die RTDB schreiben zu können.

Abbildung 4.8 zeigt die RTDB, eingehende Datenpakete sowie ausgehende Befehle. Außerdem zeigt die Abbildung die Anbindung an die Wissensbasis sowie die Möglichkeiten, die Daten in der RTDB aufzuzeichnen, sodass diese Daten später für Tests und Simulationen wiedergegeben werden können. Des Weiteren ist auch eine Visualisierung der Datenobjekte zur Fehleranalyse möglich. Bevor jedoch Datenpakete in

#### 4. ARCHITEKTUR ZUR MULTIMODALEN MENSCH-ROBOTER-INTERAKTION

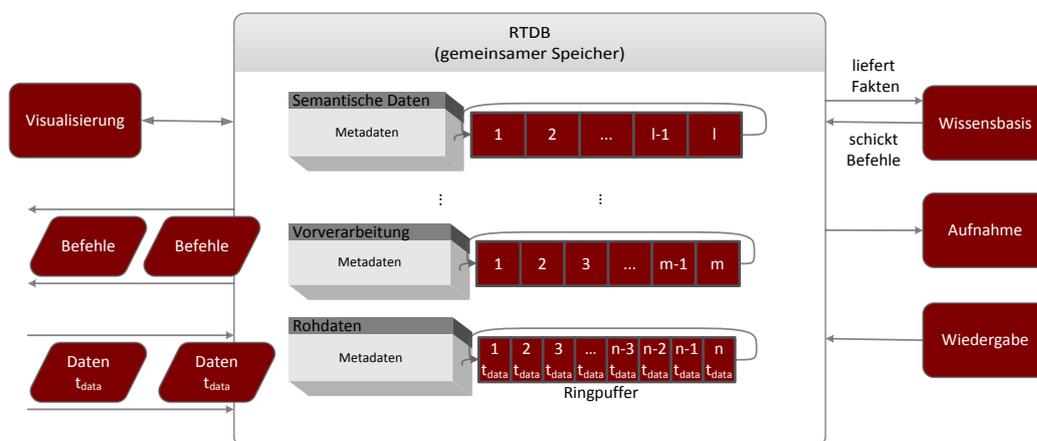


Abbildung 4.8: Datenverwaltung und Speicherung in der RTDB, vergleiche [88]

die RTDB geschrieben werden können, müssen zuerst die zugehörigen Datenstrukturen angelegt werden. Dabei muss festgelegt werden, welche Daten (beispielsweise drei Bytes für die Farbkanäle rot, grün und blau) und in welcher Größe (beispielsweise Auflösung des Bilds  $640 \times 480$ ) diese gespeichert werden sollen. Des Weiteren können die Daten in einem Ringpuffer vorgehalten werden. Neben der Aktualisierungsrate (beispielsweise 30 Hz für Bilder) muss auch die Größe und somit auch die Länge der Datenhistorie berücksichtigt werden. Für zwei Sekunden gepufferte RGB-Farbbilddaten mit 30 Hz sind folglich 60 Speicherplätze mit je  $3 \cdot 8 \cdot 640 \cdot 480 = 7\,200$  KB und somit insgesamt circa 422 MB notwendig. Neben diesen für die Datenspeicherung benötigten Informationen werden von der RTDB vorgegebene Verwaltungsdaten mit in den Metadaten der Objekte abgelegt. Die eingehenden Rohdatenpakete besitzen bereits einen Zeitstempel der Datenerstellung  $t_{data}$ . Die weiterverarbeitenden Prozesse in der RTDB können durch den gemeinsamen Speicher auf die Rohdaten im Ringpuffer zugreifen. Da der Zeitstempel der Datenerstellung  $t_{data}$  mitgeführt wird, kann nach in der Zeit zurückliegenden Daten gesucht werden. Diese Datenhistorie ist sehr hilfreich, wenn gleitende Fenster bei der Datenverarbeitung eingesetzt werden. Die aufbereiteten Daten werden anschließend wieder in die RTDB zurückgeschrieben. Die Weiterverarbeitung durch weitere Prozesse mündet schließlich in semantisch relevanten Daten, wie beispielsweise erkannte Objekte oder identifizierte Personen.

Diese Daten werden dann als Fakten an die Wissensbasis zur Verarbeitung geliefert. Durch die Verarbeitung mithilfe von Regeln in der Wissensbasis werden Befehle generiert, die über die RTDB an die Aktoren weitergeleitet werden.

## 4.5 Zusammenfassung

Die präsentierte Architektur unterstützt durch die Kombination von RTDB und ICE sowohl eine realzeitfähige Datenbereitstellung als auch eine dienstorientierte Netzwerkkommunikation zwischen den Sensoren, den Aktoren und der wissensbasierten Kontrollinstanz. Mit dem Ringpuffer der RTDB ist zudem ein Kurzzeitspeicher integriert, der die Daten für Algorithmen, die mit gleitenden Fenstern arbeiten, vorhalten kann. Der geteilte Speicher der RTDB ermöglicht zudem eine effiziente lokale Interprozesskommunikation.

Die Ablaufsteuerung wurde in der Architektur durch eine wissensbasierte Kontrollinstanz realisiert. Ein wesentlicher Vorteil dieser regelbasierten Verarbeitung ist laut [143] die Einfachheit der Implementierung. Zudem kann die Wissensbasis zur Laufzeit durch weitere Regeln ergänzt werden. Diese Fähigkeit wurde beim Lernen anhand von Instruktionen in Abschnitt 6.2 verwendet. Die Anwendung von Regeln ist immer auf die dafür notwendigerweise vorhandenen Fakten zurückzuführen. Der Ablauf ist dadurch gut erklärbar. Außerdem funktioniert laut [144] die regelbasierte Wissensverarbeitung ähnlich wie beim Menschen und ist somit gut nachvollziehbar. Ein Nachteil dieser regelbasierten Systeme ist die fehlende Verarbeitungsmöglichkeit von Fakten mit Wahrscheinlichkeiten. Hierfür kann die Wahrscheinlichkeit von Fakten mitgeführt werden [145]. Für die Entscheidungsfindung muss jedoch irgendwann auf die booleschen Werte *wahr* oder *falsch* abgebildet werden, um eine Aktion auszuführen oder nicht. Bei der Modellierung mit Wahrscheinlichkeiten wird diese Entscheidung üblicherweise durch einen Schwellwert realisiert.

Mithilfe der präsentierten multimodalen Architektur wurden die Interaktionskanäle aus Abschnitt 4.3 umgesetzt. Die entwickelte Architektur wurde prototypisch sowohl im industriellen (siehe Abschnitt 7.3) als auch im sozialen Umfeld (siehe Abschnitt 7.2) angewendet.

#### 4. ARCHITEKTUR ZUR MULTIMODALEN MENSCH-ROBOTER-INTERAKTION

---

## Blicksteuerung in Dialogsituationen

Dieses Kapitel präsentiert eine entwickelte und prototypisch realisierte Blicksteuerung für Dialogsituationen. Hierzu werden zunächst das menschliche Blickverhalten im Allgemeinen sowie relevante Größen speziell für Dialogsituationen näher erläutert. Im Anschluss daran folgt eine Übersicht über verwandte Arbeiten zu diesem Themengebiet. Danach wird das prinzipielle Konzept für das Blickverhalten sowie die dafür notwendigen Beobachtungen und Verfahren zur Extraktion dieser Beobachtungen präsentiert. Das Kapitel schließt mit Ergebnissen zu dem entwickelten Konzept sowie mit einer Zusammenfassung.

### 5.1 Motivation

Nach [146] ist der Blick neben Emotionen und Gesten ein wichtiger Aspekt der non-verbalen menschlichen Kommunikation. Der Blick dient laut [147] unter anderem dazu, den Interaktionsfluss zu regeln, Vertrautheit auszudrücken und soziale Signale wie Dominanz oder Schüchternheit zu kommunizieren. Schüchterne Personen meiden eher den direkten Blickkontakt und schauen verlegen zur Seite, während starke Persönlichkeiten direkten Blickkontakt aufnehmen, um beispielsweise ihre Entschlossenheit zu unterstreichen. Außerdem kann der Blick Aufschluss über den momentanen Aufmerksamkeitsfokus geben sowie ein erstes Indiz für zukünftige Aktionen sein [148]. Der Blick kann dabei das gemeinsame Lösen von Aufgaben fördern und beispielsweise kommunizieren, wer den ersten oder den nächsten Schritt bei der Zusammenarbeit machen soll [149].

## 5. BLICKSTEUERUNG IN DIALOGSITUATIONEN

---

Ähnlich, wie Diskursmarker in der gesprochenen Sprache den Sprecherwechsel [150] organisieren können, so kann auch durch den Blick der Dialog strukturiert werden. In [151] wird durch Blick bei mehreren Interaktionspartnern deren Rolle im Dialog kommuniziert. Laut [152] beeinflusst das Blickverhalten des Dialogpartners auch die Wahrnehmung und die Bewertung des Gesprächs. Häufiger Blickkontakt führt zu einer positiveren Einschätzung des Gegenübers und des Dialogs. Durch den Blick kann Interesse am Thema oder eine Zuneigung zum Gesprächspartner kommuniziert werden.

Das menschliche Blickverhalten ist eine Kombination aus einer ziel-gesteuerten Wahrnehmung (engl. Top-Down) und der durch Reize induzierten Wahrnehmung (engl. Bottom-Up) [153]. Diese Einflüsse auf das Blickverhalten in Dialogsituationen sind ohne Anspruch auf Vollständigkeit exemplarisch in Abbildung 5.1 dargestellt.



**Abbildung 5.1:** Exemplarische Top-Down und Bottom-Up Einflüsse auf das Blickverhalten in Dialogsituationen

Laut [154] unterscheidet sich der Blick einer Person auch je nachdem, ob die Person gerade Zuhörer oder Sprecher ist. Außerdem kann für eine Top-Down Steuerung

der Blick auf im Gespräch relevante Objekte fokussiert werden, um deren Bedeutung hervorzuheben. Bei der reizinduzierten Steuerung spielt die auditive oder visuelle Salienz eine wesentliche Rolle. Die Reize stechen hierbei beispielsweise aufgrund von Intensitäts- oder Farbunterschieden, speziellen Formen, Bewegungen, Lautstärke oder unerwartetem Erscheinen aus den umgebenden Reizen heraus.

Die Wirkung von Reizen auf die Wahrnehmung kann durch dauerhafte oder wiederholte Darbietung [155] auch reduziert werden oder sogar komplett verschwinden. Außerdem gibt es beim menschlichen Auge auch Bewegungen, die auf das Aufnahme- und Verarbeitungsprinzip der Augen zurückzuführen sind. Da nur ein kleiner Teil des Auges, die Fovea, hochauflösend ist, erstellt der Mensch mittels wanderndem Blick eine hochauflösende Darstellung von Objekten beziehungsweise seinem Umfeld. Durch minimale Augenbewegungen werden die Bilder auf der Netzhaut sogar aktualisiert, wenn wir den Blick fixieren, da die Zellen im Auge auf Änderungen reagieren und ansonsten das Bild langsam verblassen würde [156].

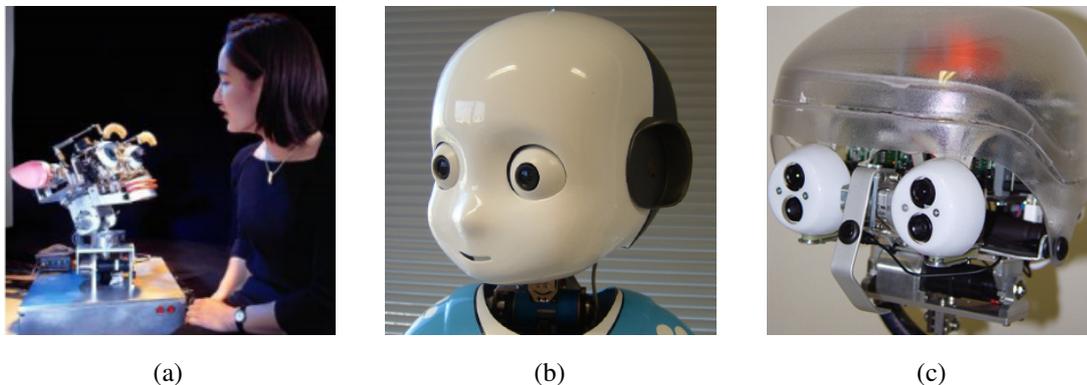
Für eine insgesamt natürlichere Interaktion mit Robotern, insbesondere im sozialen Umfeld, sollte also auch der Blick berücksichtigt werden. Für das hier entwickelte Blickmodell stand dabei die Interaktion mit dem Nutzer im Vordergrund. Deshalb wurde das Modell in diesem Abschnitt basierend auf Studien zum menschlichen Blickverhalten bezüglich Gesichtern entwickelt, da diese in der Dialogsituation einen besonders relevanten Reiz darstellen und der Roboter dadurch dem Nutzer seine momentane Aufmerksamkeit kommunizieren kann.

## 5.2 Verwandte Arbeiten

In diesem Abschnitt werden weitere Arbeiten vorgestellt, die sich ebenfalls mit Modellen für die Blicksteuerung befassen. Die entwickelten Modelle werden zur Evaluierung in Kommunikationssituationen entweder an echte Roboterköpfe oder an virtuelle Avatare angebunden.

[157] präsentiert einen Kopf mit zwei Kameraaugen, die über eine realzeitfähige Vergenzsteuerung der Augen verfügen. Dafür wird aus den zwei Bildern der Kamera die Tiefeninformation berechnet und die Augen sowie der Kopf entsprechend auf das Ziel ausgerichtet. Mittels optischem Fluss und Berücksichtigung der Eigenbewegung von Kamera und Kopf wurden damit auch sanfte Verfolgungsbewegungen (engl.: smooth pursuit) von sich bewegenden Zielen realisiert.

In [158] und [159] wird eine aktive Blicksteuerung für den Roboter Kismet aus Abbildung 5.2 (a) beschrieben. Der Roboterkopf wurde primär für die Darstellung



**Abbildung 5.2: Roboter mit Augensteuerung** - (a) Kismet zur Emotionsdarstellung von Breazeal [158], (b) der Roboterkopf von iCub [160] und (c) Kopf von ARMAR III aus Karlsruhe [161]

von Emotionen entwickelt und bietet mit 15 Freiheitsgraden genügend Möglichkeiten, die Basisemotionen aus [120] darzustellen. Der Kopf verfügt über drei Freiheitsgrade, während die Augen gemeinsam in der Vertikalen geneigt werden können und sich einzeln in der horizontalen Achse bewegen lassen. Für die Berechnung der Aufmerksamkeit für die Augensteuerung werden die drei Farbkanäle rot, grün und blau sowie eine Karte mit markierten Pixeln, die Hautfarbe enthalten, und eine weitere Karte mit Bewegungsinformationen anhand von Differenzbildern verwendet. Zusätzlich wird auch eine Gewöhnung an Reize über die Zeit mitgeführt. Die Gewichtung dieser Merkmale zur Berechnung der Aufmerksamkeit kann entsprechend der Aufgabe angepasst werden, um beispielsweise bei der Suche nach einem roten Objekt diesen Farbkanal höher zu bewerten.

[162] nutzt einen Pinguin als Kommunikationsavatar für eine Nutzerstudie mit 37 Personen. Dabei führt der Roboter ein Gespräch über Gegenstände im Raum. Die Probanden wurden in zwei Gruppen eingeteilt, wobei der Avatar bei einer Gruppe seinen Blick auf die Gegenstände im Gespräch richtet und dann wieder seinen Gesprächspartner anschaut, während in der zweiten Gruppe zu Beginn der Gesprächspartner lokalisiert wird und der Blick dann nicht mehr von diesem weicht. Die Studie zeigt, dass, wenn der Avatar seinen Blick aktiv im Gespräch einsetzt, die Probanden sich länger mit dem Avatar beschäftigen und das Gespräch als interessanter empfinden sowie dadurch auch ihr Blickverhalten beeinflussen lassen.

In [160] wird ein Modell für die auf Salienz basierende Steuerung der Aufmerksamkeit für den Roboter iCub präsentiert. Der Roboter erfasst die Umwelt über sei-

ne Roboteraugen, die in Teil (b) von Abbildung 5.2 zu sehen sind. Dabei erreicht er ein sehr gutes Blickverhalten, das lediglich auf reizinduzierte Salienz ausgerichtet ist. Die Erfassung der Umwelt entspricht einem menschenähnlichen Prinzip. Das Bild der Umwelt wird über die in die Roboteraugen eingebauten Kameras Stück für Stück zusammengesetzt, wobei dann eine visuelle Kopie der Umgebung im System generiert wird.

Der Roboter ARMAR III aus Karlsruhe [161] ist ein lernfähiger Roboter, der für den Einsatz in der Küche entwickelt wurde. Der Roboter verfügt über einen Kopf und Augen, um seiner Umgebung den momentanen Aufmerksamkeitsfokus der Bildverarbeitungsalgorithmen mitteilen zu können. In die Augen sind insgesamt vier Kameras integriert. In jedes Auge zwei, wobei je eine für den Nah- und eine für den Fernbereich genutzt wird.

Auch für virtuelle Gesichter beziehungsweise digitale Avatare wird an Blickmodellen geforscht. Der Blick ist auch hier ein wichtiges Element für eine glaubwürdige Darstellung der Charaktere, weshalb die Animationen häufig noch manuell erstellt werden. Aufgrund immer häufigerer virtueller Charaktere wird jedoch auch an Modellen geforscht, die eine automatische Animation erlauben. Dafür wurde in [163] ein Modell für Sakkaden entwickelt, das auf Blickerfassungsergebnissen beruht. Anschließend wurde das Modell gegenüber zufälligen Sakkadenbewegungen subjektiv durch Probanden bewertet. In [164] wurde basierend auf Blickstudien ein Modell für dialogorientierte Agenten (engl. conversational agents) generiert. Dafür wurde ein Bayesisches Netz erstellt, das den vorherigen Zustand von Sprecher und Zuhörer nutzt, um damit den nachfolgenden zu steuern. Dieses Netz wurde aus einer Kombination der statistisch ermittelten Blickdaten der Studie sowie einem funktionalen Modell erstellt. [165] versucht, ein ausdrucksstarkes Blickmodell zu erstellen, um mit den Augen auch Emotionen auszudrücken. Dabei werden neben den Augen auch der Kopf sowie der Torso in die Steuerung miteinbezogen, um eine insgesamt realistischere Darstellung zu erreichen. Allerdings beschäftigt sich das Modell nicht damit, wohin der Blick gerichtet werden soll, sondern mit der Art und Weise der Animation zu einem gegebenen Ziel und einer gegebenen Emotion.

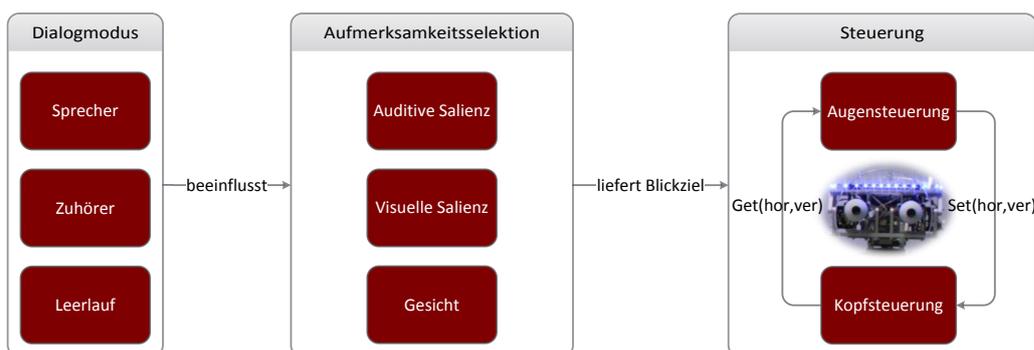
In der Studie von [128] wird das Blickverhalten zwischen zwei Menschen in Dialogsituationen mittels Augentracking untersucht. Danach wurde eine erste Analyse der Daten durchgeführt, die in nachfolgenden Arbeiten für die Erstellung eines Blickmodells verwendet werden soll. Interessanterweise wurden bei den Untersuchungen relativ wenige direkte Blickkontakte festgestellt. Für die Analyse wurden die Blickdaten der Probanden sowie die Umgebungsbilder mitgespeichert und anschließend mittels

der Software FaceAPI [129] die Gesichtsmerkmale extrahiert, um dann Rückschlüsse auf die Häufigkeiten der betrachteten Merkmale ziehen zu können.

### 5.3 Modell für Blickverhalten

Abbildung 5.3 zeigt die konzeptionelle Idee für die Steuerung des Blickverhaltens. Der Roboter nimmt entweder als *Sprecher* oder *Zuhörer* am Dialog teil. Findet aktuell kein Dialog statt, befindet sich das System im Dialogmodus *Leerlauf* und wartet auf den nächsten Dialogpartner. Die Aufmerksamkeitsselektion wird dabei von dem Dialogmodus, in dem sich der Roboter gerade befindet, beeinflusst. Die Aufmerksamkeitsselektion ist die Komponente, die aus den möglichen Beobachtungen visuelle Salienz, auditive Salienz und Gesicht eine Beobachtung selektiert. Das Gesicht gehört zwar streng genommen ebenfalls zu den visuellen Salienzen, wird jedoch aufgrund der besonderen Bedeutung im Dialog in dieser Arbeit getrennt davon betrachtet. Zur Abgrenzung dieser beiden Reize gegenüber der auditiven Salienz werden Gesicht und visuelle Salienz in dem Begriff *visuelle Information* zusammengefasst.

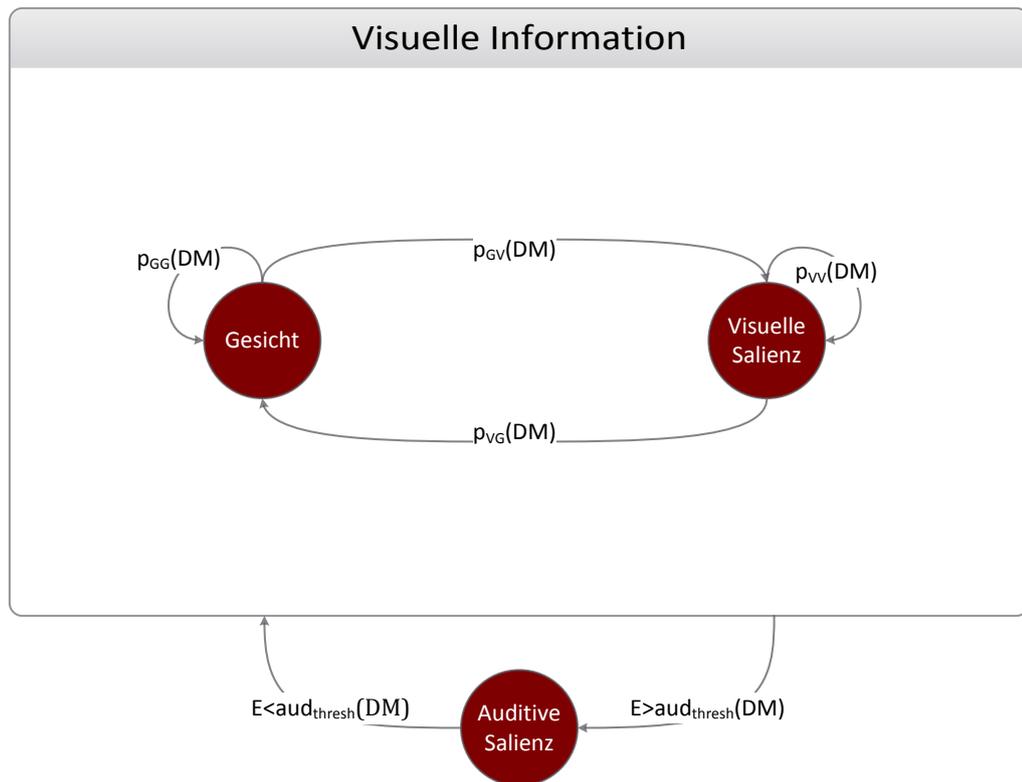
Die Aufmerksamkeitsselektion bereitet die selektierte Beobachtung weiter auf, um diese als Blickziel für die Blicksteuerung bereitstellen zu können. Für das selektierte Blickziel muss dann die Ansteuerung der beiden Augen sowie die Nachführung des Kopfs berechnet werden.



**Abbildung 5.3:** Konzept für die Steuerung des Blickverhaltens

Als Modell für die Aufmerksamkeitsselektion wurde ein stochastischer Zustandsautomat gewählt. Dieser Zustandsautomat modelliert die drei Beobachtungen als Zustände und die Transitionen zwischen diesen Zuständen hängen von Ereignissen und

Wahrscheinlichkeiten ab. Um den Einfluss des Dialogmodus in das Modell miteinzubeziehen, sind die Übergangswahrscheinlichkeiten zwischen den Zuständen nicht fest definiert, sondern in Abhängigkeit des aktuellen Dialogmodus zu gestalten. Dieses Blickmodell ist in Abbildung 5.4 zu sehen.



**Abbildung 5.4:** Blickmodell für die Übergänge zwischen den Beobachtungen

Gesichter und visuelle Salienz werden im Teilautomaten mit dem Bezeichner *visuelle Information* zusammengefasst. Die Wahrscheinlichkeiten der Transitionen zwischen den Zuständen in diesem Teilautomaten werden als Funktion des Dialogmodus mit  $p(DM)$  modelliert. Bei auditiven Salienzen gilt eine besondere Bedingung für die Transition. Für die Transition aus dem Teilautomaten *visuelle Information* muss die Audiosignalenergie  $E$  über einem Schwellwert  $aud_{thresh}$  liegen, der wiederum davon abhängt, wie der aktuelle Dialogmodus lautet, und der folglich als Funktion  $aud_{thresh}(DM)$  dargestellt wird. Die auditive Salienz bleibt solange Blickziel, wie die Audiosignalenergie über diesem Schwellwert liegt. Die Rücktransition erfolgt in den Zustand, der zuvor im Teilautomaten *visuelle Information* aktiv war.

Für die Steuerung der Augen wurde neben den Sakkaden, die im Prinzip sehr schnelle Sprünge der Augen zwischen den Beobachtungen sind, auch eine Verfolgung von bewegten Reizen implementiert. Für eine harmonische Steuerung von Kopf und Augen wurde eine Studie analysiert, die Werte über den Kopf- und Augenversatz bei den Probanden liefert.

Da diese Werte bei den Probanden von [166] für die Kopfnachführung unterschiedlich waren, wurde für die folgende Arbeit der Mittelwert herangezogen. Für das Blickmodell wird somit der Wert von  $15^\circ$  als Schwellwert für eine Nachregelung des Kopfes in der horizontalen Bewegung verwendet. Der Kopf wird in diesem Fall auf den neuen Blickpunkt eingeregelt und die Information über den Kopfwinkel genutzt, um die Augen in die entgegengesetzte Richtung wandern und somit den Blick auf dem Ziel verweilen zu lassen. Für die vertikale Achse wurde ein Schwellwert von  $10^\circ$  gewählt, da hier der Roboterkopf weniger Bewegungsspielraum hat und dazu keine Ergebnisse aus der Studie hervorgehen.

### 5.4 Relevante Beobachtungen

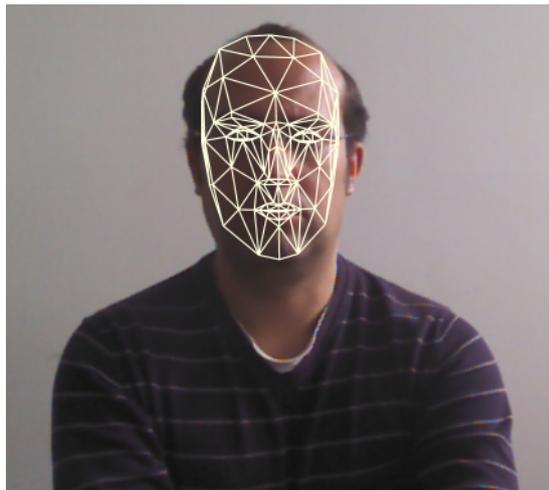
Wie bereits erläutert, sind einige Beobachtungen für die Blicksteuerung in Dialogsituationen besonders interessant. Für die Umsetzung des menschlichen Blickverhaltens im Bezug auf Gesichter müssen zunächst die Gesichtsmerkmale des Dialogpartners extrahiert werden.

#### 5.4.1 Gesichtsmerkmale

Für die Gesichtsmerkmale wird das Microsoft Kinect für Windows SDK verwendet, das auch eine Gesichtsverfolgung bereitstellt. Dabei nutzt die Software die Farb- und Tiefenbilder und das CANDIDE-3-Modell [167]. Das CANDIDE-3-Modell ist die dritte Iteration für ein dreidimensionales Drahtgittermodell zur Modellierung eines Gesichts. Das CANDIDE-1-Modell verfügte über 79 Punkte und 108 Oberflächen, der Nachfolger CANDIDE-2-über 160 Punkte und 238 Oberflächen. Das CANDIDE-3-Modell wurde schließlich an die Merkmalspunkte zur Gesichtsanimation des Moving Pictures Expert Group (MPEG)-4 Standards [168] angepasst.

Die Gesichtsverfolgung des Microsoft Kinect für Windows SDK liefert neben den Punkten für die Gesichtsmerkmale auch sechs Animationseinheiten (engl.: Animation Units) und elf Formeinheiten (engl.: Shape Units) des Gesichtsmodells. Zusätzlich zu diesen Informationen wird auch der aktuelle Status der Verfolgung sowie die Position

des Gesichts und die vermutete Kopfpose in den drei Winkeln  $(\alpha, \beta, \gamma)$  mitgeliefert. Für eine erfolgreiche Verfolgung des Gesichts sollte der Kopf nicht mehr als  $20^\circ$  vertikal geneigt sein. Horizontal kann der Kopf bis zu  $45^\circ$  verdreht sein und sollte nicht mehr als  $90^\circ$  gekippt sein, damit die Gesichtsverfolgung erfolgreich funktioniert, wie in Abbildung 5.5 zu sehen ist.



**Abbildung 5.5:** Parametrisiertes CANDIDE-3-Gesichtsmodell [167], berechnet mittels Microsoft Kinect für Windows Gesichtsverfolgungs-SDK

Mithilfe der gelieferten Gesichtsmerkmale können weitere Informationen extrahiert werden. So kann anhand der Variation der Merkmale im Mundbereich die Aktivität des Dialogpartners abgeleitet werden, um beispielsweise diesem Gesichtsbereich eine höhere Gewichtung im Blickmodell zu geben, wenn der Gesprächspartner gerade spricht. Die aktuelle Rolle des Gesprächspartners sowie die Rolle des Roboters sind ebenso relevant für die Blicksteuerung. Die Information über die momentane Rolle des Roboters und somit auch über die Rolle des Gesprächspartners kann vom Dialogmanager des Systems abgeleitet werden.

Als Ergänzung zu diesen auf dem Gesichtsmodell basierenden Informationen sind die auditive und die visuelle Salienz für die weitere Verwendung im Blickmodell zu bestimmen, welche laut [169] und [170] die Blicksteuerung wesentlich beeinflussen. Die Berechnung dieser Salienzen wird in den folgenden Abschnitten näher erläutert.

### 5.4.2 Auditive Salienz

Der Wert für die auditive Salienz wurde über die Energie eines gefensterten Audiosignals berechnet und der zugehörige Winkel der Audioquelle in der Azimutalebene

## 5. BLICKSTEUERUNG IN DIALOGSITUATIONEN

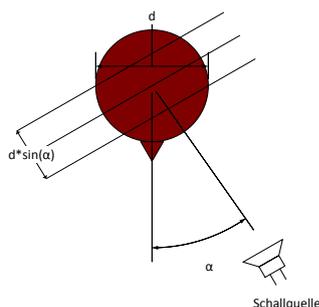
---

ne (horizontale Ebene) mittels interauraler Laufzeitdifferenzen (engl.: Interaural Time Difference (ITD)) bestimmt. Das Prinzip der Schallquellenermittlung basiert darauf, dass das emittierte Signal zu unterschiedlichen Zeitpunkten bei den Ohren eintrifft, falls das Signal nicht direkt vor oder hinter dem Kopf emittiert wird.

Dazu wird durch Kreuzkorrelation das Maximum zwischen dem linken und dem rechten Audiosignal ermittelt. Dies erlaubt zusammen mit der Abtastfrequenz Rückschlüsse darauf, wie groß die interaurale Zeitdifferenz ist:

$$t_{ITD} = \frac{n}{f} \quad (5.1)$$

Dabei ist  $n$  der Unterschied zwischen den Maxima des linken und rechten Audiosignals, ausgedrückt in der Anzahl der Rahmen zwischen den Signalen und  $f$  die Abtastfrequenz.



**Abbildung 5.6:** Interaurale Zeitdifferenz

Kennt man nun den Abstand  $d$  zwischen den Ohren beziehungsweise den Mikrofonen, so kann man auf den Winkel folgendermaßen zurückschließen:

$$\Delta s = d \cdot \sin(\alpha) \quad (5.2)$$

Hierbei ist  $\Delta s$  die Strecke, die den Laufzeitunterschied verursacht und mit der Geschwindigkeit  $c$  in der Zeit  $t_{ITD}$  zurückgelegt wird.

Daraus ergibt sich der gesuchte Winkel zu:

$$\alpha = \arcsin\left(\frac{nc}{fd}\right) \quad (5.3)$$

Die visuell bestimmte Gesichtsposition des Sprechers kann außerdem zur Unterscheidung zwischen Audiosignalen des Gesprächspartners und anderen Geräuschquellen herangezogen werden.

### 5.4.3 Visuelle Salienz

Für die Berechnung der visuellen Salienz wurde auf die Kinect-Sensorbilder eine schnelle Implementierung des Saliency Using Natural statistics (SUN)-Algorithmus aus [171] angewendet. Der deshalb FastSUN genannte Algorithmus versucht, ein menschenähnliches Suchverhalten, trainiert auf Bildsequenzen, nachzubilden. Dabei wird der Ansatz verwendet, dass ein Objekt im Bild gesucht wird, dessen Klassenzugehörigkeit für einen Punkt  $\mathbf{m}$  mit der Zufallsvariable (ZV)  $c_m$  angegeben wird. Die ZV  $l_m$  beschreibt die Position und die ZV  $f_m$  die Merkmale des Punktes  $\mathbf{m}$ . Damit kann die visuelle Salienz  $s_m$  am Punkt  $\mathbf{m}$  folgendermaßen definiert werden:

$$s_m = \frac{p(f_m, l_m | C = 1)p(C = 1)}{p(f_m, l_m)} \quad (5.4)$$

Mit der Annahme, dass Merkmale und Position unabhängig sind und bedingt unabhängig bei gegebener Klassenzugehörigkeit  $C = 1$  und Logarithmierung, ergibt sich aus 5.4:

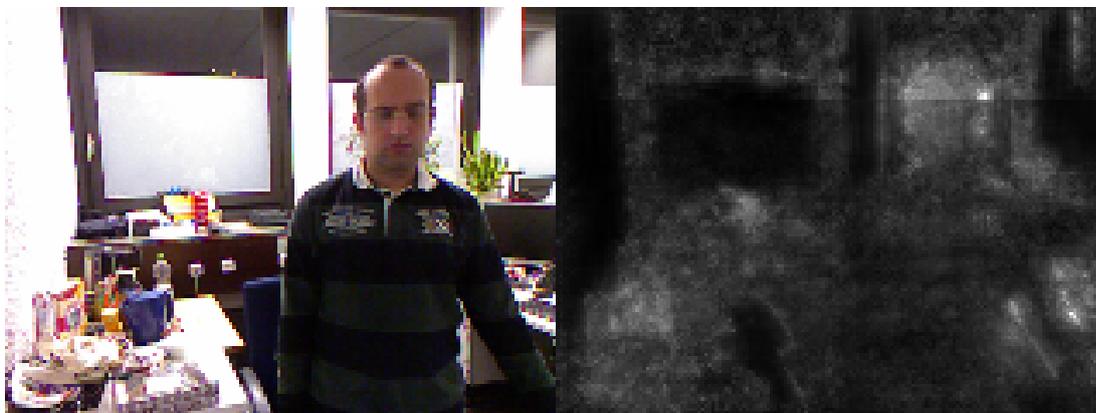
$$\log s_m = -\log p(f_m) + \log p(f_m | C = 1) + \log p(C = 1 | l_m) \quad (5.5)$$

Der erste Summand aus 5.5 beschreibt die Salienz für seltene Merkmale im Bild, der zweite Summand beschreibt die Wahrscheinlichkeit für Merkmale für eine gegebene Klassenzugehörigkeit und der letzte Summand gibt das Vorwissen für die Auftrittswahrscheinlichkeit der Klasse bei gegebener Position an. Werden keine speziellen Objekte im Bild gesucht, reduziert sich die Gleichung auf den ersten Summand. Das bedeutet, dass Merkmalen, die im Bild eher selten sind, ein hoher Wert für die Aufmerksamkeit zugewiesen wird und häufig auftretende Merkmale niedrig bewertet werden. Dies entspricht auch der intuitiven Definition eines Reizes. Nun kann für jeden Punkt  $\mathbf{m}$  eines Bilds  $I_{x,y}$  der zugehörige Wert der Salienz  $\log s_m$  berechnet werden, was zu einer wie in Abbildung 5.7 dargestellten Salienzkarte führt. Die Salienz ist dabei mit Graustufen codiert, wobei hohe Werte heller und niedrige Werte dunkler dargestellt werden. Man sieht, dass gleichfarbige Flächen eher dunkel erscheinen und die bunten Objekte auf dem Tisch oder am Fenster hellere Werte in der Karte erzeugen.

Aus dieser Karte kann nun der hellste Punkt als Kandidat für die visuelle Salienz bestimmt werden:

$$\arg \max_{m \in I_{x,y}} \log s_m \quad (5.6)$$

Der Vollständigkeit halber sei erwähnt, dass der Algorithmus über weitere Parameter konfiguriert werden kann. Für die Kontrastberechnung kann entweder die Intensität eines Graustufenbildes oder die Rot-, Grün- und Blau-Farbkanalinformation des Bilds

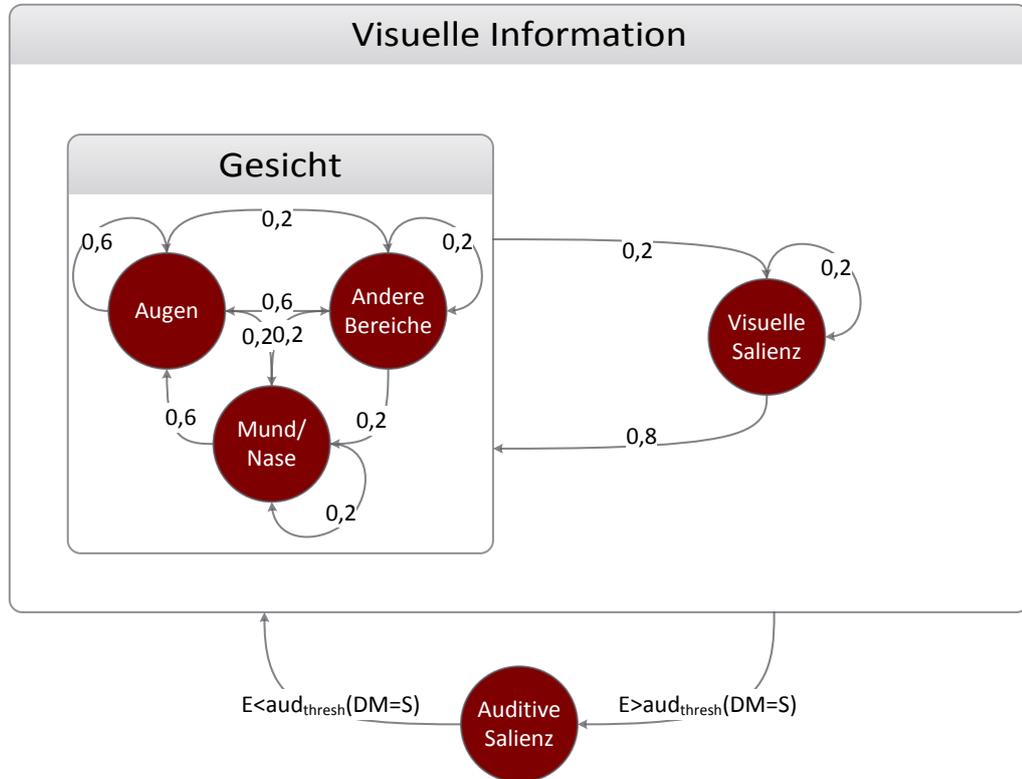


**Abbildung 5.7:** Farbbild und zugehörige Salienzkarte skaliert auf  $160 \times 120$  Pixel

verwendet werden. Zeitliche Veränderungen der Salienzen können berücksichtigt werden, um beispielsweise Bewegungen im Bild bei der Salienzberechnung eine höhere Gewichtung geben zu können. Des Weiteren kann eingestellt werden, wie schnell die Salienzen mit der Zeit abklingen, um die Gewöhnung an den Reiz zu simulieren. Die Rechenzeit wird von der verwendeten Ausgangsbildgröße für den Algorithmus dominiert. In dieser Arbeit wurde das Bild auf  $160 \times 120$  Pixel skaliert und dabei eine mittlere Laufzeit von 110 ms gemessen. Um Rauschen bei der Salienzberechnung zu reduzieren, wurde eine zeitliche Glättung genutzt, um kurze starke Salienzen herauszufiltern, die zu einem sehr unruhigen Blick geführt hätten. Da beim Kinect-Sensor auch die Tiefeninformation zu den Pixeln vorhanden ist, kann außerdem die Tiefe zur Abschwächung der Reize mit wachsender Distanz zum Betrachter verwendet werden. Damit werden nähere Reize bei der Berechnung der Salienz höher gewichtet.

### 5.5 Implementierung und Ergebnisse

Mit der Architektur aus Kapitel 4.4 wurde auf der Forschungsplattform Enhanced Living ASsistant (ELIAS) (siehe Abschnitt 7.1.1) eine prototypische Implementierung des Blickmodells umgesetzt. Hierfür wurden insbesondere die Blickmessungen aus dem Experiment 2 von [172] und [173] verwendet, die ergeben haben, dass der Mensch zu ungefähr 60% der Zeit seinen Blick auf die Augen, zu 20% auf die Mund- und Nasenregion und zu 20% auf den Rest des Gesichts richtet. Das Blickmodell aus Abschnitt 5.3 wurde dazu um einen weiteren Teilautomaten für die Gesichtsmerkmale erweitert und ist für den Dialogmodus Sprecher mit den Transitionswahrscheinlichkeiten in Abbildung 5.8 visualisiert.



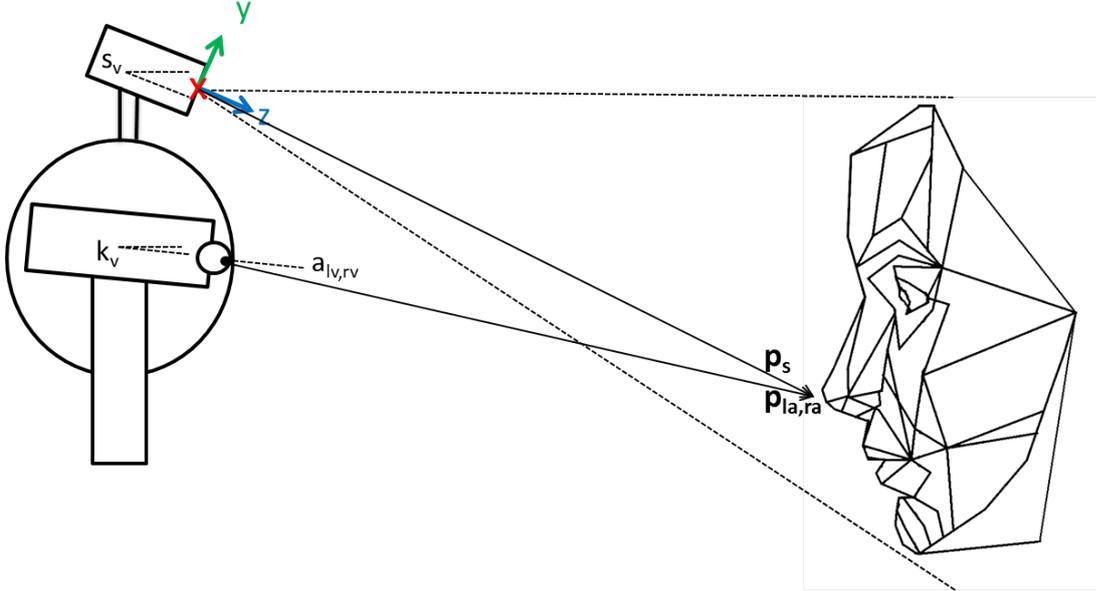
**Abbildung 5.8:** Blickmodell mit den Transitionswahrscheinlichkeiten zwischen den relevanten Beobachtungen für den Dialogmodus Sprecher

Wie bereits zuvor erwähnt, steht dabei  $E$  für die Energie der auditiven Salienz und  $\text{aud}_{\text{thresh}}(DM = S)$  für einen Schwellwert im Dialogmodus Sprecher, ab dem auf den auditiven Reiz reagiert wird. Für die Implementierung wurde  $\text{aud}_{\text{thresh}}(DM = S)$  als Zehnfaches der Umgebungslautstärke bestimmt, um somit die Transition nur bei starken Reizen zu schalten.

Während die Winkel der auditiven Salienz unter Berücksichtigung der horizontalen Rotation des Roboterkopfes als Winkel für die horizontale Stellung der Augen verwendet werden können, müssen die visuelle Salienz sowie die Gesichtsmerkmale noch in Winkel für das linke und das rechte Auge umgerechnet werden.

Abbildung 5.9 zeigt hierfür zur Veranschaulichung links schematisch den Roboterkopf samt Augen und Sensor und auf der rechten Bildseite ein Gesichtsmodell.

Die selektierte visuelle Information (Gesichtsmerkmale oder visuelle Salienz) liegt zunächst als Punkt  $\mathbf{p}_s$  im Koordinatensystem des Sensors vor. Dieser Punkt muss nun



**Abbildung 5.9:** Sensor, Roboterkopf und -augen sowie Gesichtsmodell in der Seitenansicht [174]

für die Ansteuerung der Roboteraugen in Punkte des linken  $\mathbf{p}_{la} = (x_{p_{la}}, y_{p_{la}}, z_{p_{la}})^T$  beziehungsweise rechten  $\mathbf{p}_{ra} = (x_{p_{ra}}, y_{p_{ra}}, z_{p_{ra}})^T$  Augenkoordinatensystems überführt werden. Im Anschluss daran können dann die benötigten vertikalen Winkel  $a_{lv}$  und horizontalen Winkel  $a_{lh}$  für das linke Auge sowie  $a_{rv}$  und  $a_{rh}$  für das rechte Auge mithilfe der folgenden zwei Gleichungen berechnet werden:

$$\mathbf{p}_{la} = \mathbf{t}_{la} + R_{k_h, k_v}(\mathbf{t}_k + R_{s_h, s_v}(\mathbf{t}_s + \mathbf{p}_s)) \quad (5.7)$$

$$\mathbf{p}_{ra} = \mathbf{t}_{ra} + R_{k_h, k_v}(\mathbf{t}_k + R_{s_h, s_v}(\mathbf{t}_s + \mathbf{p}_s)) \quad (5.8)$$

Mithilfe der Gleichungen werden die Punkte zunächst um  $\mathbf{t}_s$  ins Sensorzentrum verschoben und um die Winkel  $s_h$  und  $s_v$  des Sensors rotiert mit der Matrix  $R_{s_h, s_v}$ . Anschließend folgt eine Translation ins Roboterkopffzentrum anhand von  $\mathbf{t}_k$  sowie einer Rotation um die Winkel  $k_h$  und  $k_v$  mit der Matrix  $R_{k_h, k_v}$ . Danach müssen die Punkte ins linke  $\mathbf{t}_{la}$  beziehungsweise rechte  $\mathbf{t}_{ra}$  Augenzentrum verschoben werden.

Die benötigten Winkel  $a_{lv}$ ,  $a_{lh}$ ,  $a_{rv}$ ,  $a_{rh}$  zur Augensteuerung lassen sich aus den Punkten  $\mathbf{p}_{la}$  und  $\mathbf{p}_{ra}$  folgendermaßen berechnen:

$$a_{lv} = \arctan\left(\frac{y_{pla}}{z_{pla}}\right) \quad (5.9)$$

$$a_{rv} = a_{lv} \quad (5.10)$$

$$a_{lh} = \arctan\left(\frac{x_{pla}}{z_{pla}}\right) \quad (5.11)$$

$$a_{rh} = \arctan\left(\frac{x_{pra}}{z_{pra}}\right) \quad (5.12)$$

Da sich  $\mathbf{p}_{la}$  und  $\mathbf{p}_{ra}$  lediglich in der  $x$ -Komponente des Vektors unterscheiden, gilt  $a_{lv} = a_{rv}$ .

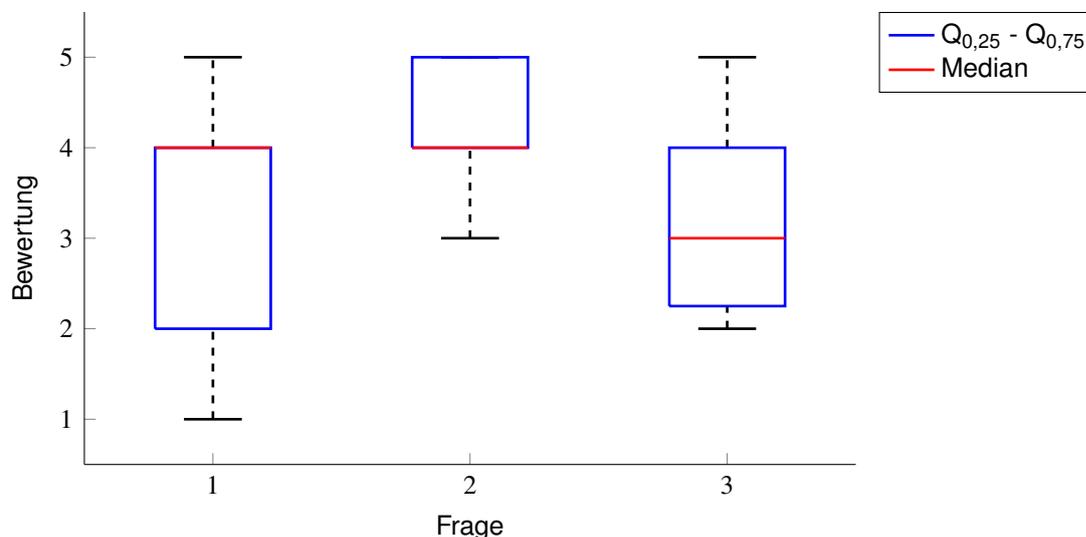
### 5.5.1 Ergebnisse

Die Bewertung der Natürlichkeit des Blickverhaltens ist mit objektiven Mitteln kaum möglich. Deshalb wurden für eine Beurteilung des Blickverhaltens im Rahmen eines kurzen Dialogszenarios die subjektiven Eindrücke von Probanden anhand eines Fragenkatalogs näher untersucht. An dem Versuch haben insgesamt 20 Personen teilgenommen, von denen 18 männlichen und zwei weiblichen Geschlechts waren. Da die Versuchspersonen ausschließlich Studenten waren, lag das Alter der Probanden zwischen 22 und 26 Jahren. Hierfür wurde ein circa 5 Minuten langer Dialog geführt. Der Roboter hat sich zunächst selbst kurz vorgestellt und danach einige Fragen an den Probanden gestellt. Dabei wurden die Teilnehmer beispielsweise gefragt, wie alt sie sind, was sie studieren etc. Nach dem Dialog hatten die Probanden die Möglichkeit, die Reaktion des Roboters auf visuelle und auditive Saliens ohne Dialogsituation zu testen. Der Roboter befindet sich dabei im Dialogmodus *Leerlauf*. Die Probanden konnten nun beispielsweise durch Winken, Klatschen oder Vorbeigehen an der Plattform das resultierende Blickverhalten studieren.

Der Fragenkatalog für die Probanden wurde in vier Bereiche aufgeteilt. Die ersten drei Bereiche orientieren sich an den drei Beobachtungen aus Abschnitt 5.4. Der vierte und letzte Teil behandelt allgemeine Fragen zur Interaktion. Die subjektive Bewertung der Interaktion durch die Probanden erfolgte mittels einer Likert-Skala [175] von 1 - *stimme nicht zu* bis 5 - *stimme voll zu*. Im Folgenden werden die Fragen zu dem jeweiligen Bereich präsentiert und die Ergebnisse zu diesen Fragen werden in Form von Boxgraphen dargestellt.

## 5. BLICKSTEUERUNG IN DIALOGSITUATIONEN

---

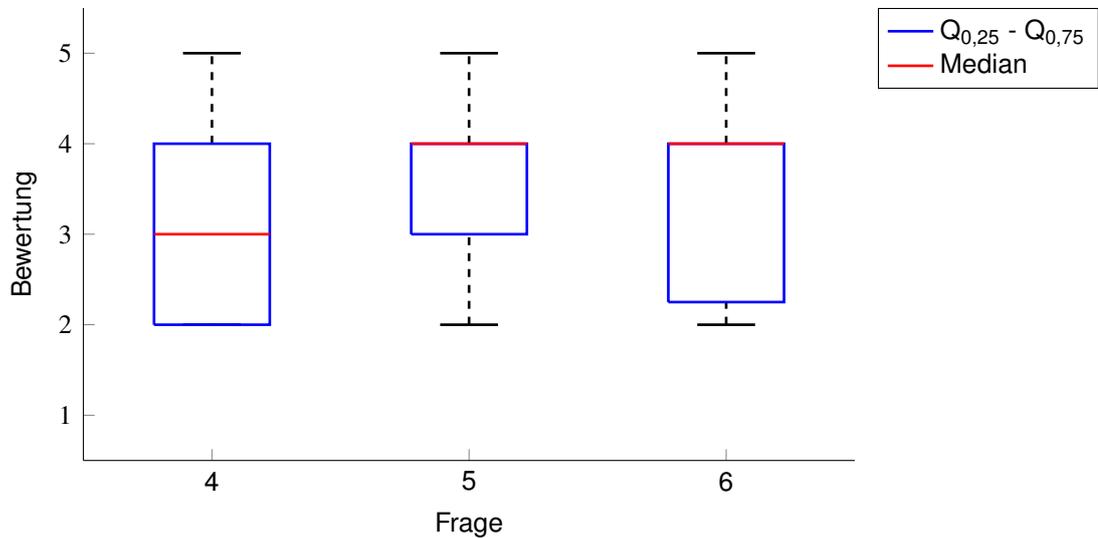


**Abbildung 5.10: Ergebnisse zu den Fragen bezüglich der Gesichtsmerkmale** - Diese Abbildung zeigt das Evaluierungsergebnis der Probanden zum Block Gesichtsmerkmale.

### Fragenblock: Gesichtsmerkmale

- 1 - Ich hatte den Eindruck, einen Dialog mit einem Menschen zu führen.**
- 2 - Mir sind die Sakkaden während des Dialogs aufgefallen.**
- 3 - Der Roboter hat mich schnell erkannt.**

Die erste Frage aus Abbildung 5.10 zielt auf die Natürlichkeit der Interaktion ab. Hier liegt der Median bei vier, somit ist die Interaktion überdurchschnittlich gut bewertet. Allerdings ist auch zu sehen, dass die Werte das gesamte Bewertungsspektrum einnehmen und somit nicht bei allen Probanden ein guter Eindruck erzielt werden konnte. Außerdem liegen 50% der Bewertungen zwischen zwei und vier. Die zweite Frage wurde von allen Probanden überdurchschnittlich bewertet, somit wurden die schnellen Augenbewegungen zwischen den erkannten Gesichtsmerkmalen auch gut von den Probanden wahrgenommen. Die letzte Frage zielt auf die Erkennungsgeschwindigkeit der Gesichtsmerkmale ab. Hier zeigt der Median von drei noch einen klaren Bedarf zur Verbesserung der Initialisierungsgeschwindigkeit. Abhilfe könnte hier auch ein schnellerer Prozessor schaffen.



**Abbildung 5.11: Ergebnisse zur visuellen Salienz** - Diese Abbildung zeigt das Evaluationsergebnis der Probanden zur visuellen Salienz.

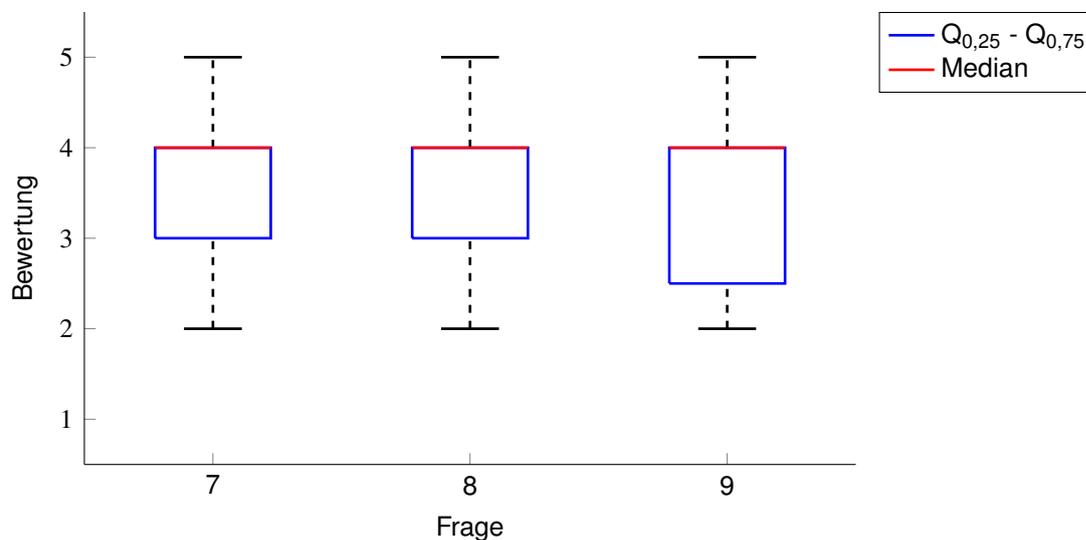
#### **Fragenblock: Visuelle Salienz**

**4 - Die Reaktion des Roboters auf visuelle Reize mit Dialog war realistisch.**

**5 - Die Reaktion des Roboters auf visuelle Reize ohne Dialog war realistisch.**

**6 - Der Roboter hat mich schnell wiedererkannt, wenn ich wieder vor ihm stand.**

Die Antworten zu den Fragen der visuellen Salienz sind in Abbildung 5.11 zu sehen. Anhand des Vergleichs von Frage vier und fünf ist zu sehen, dass die Reaktion des Roboters auf visuelle Reize außerhalb des Dialogszenarios von den Probanden als realistischer bewertet wurde. Dies liegt wahrscheinlich daran, dass ohne Dialog bei der Augensteuerung die Gesichtsmerkmale nicht berücksichtigt werden und somit nur auf visuelle und auditive Salienzen direkt reagiert wird. Im Dialogmodus wird aufgrund des Modells nicht zwangsweise auf einen visuellen Reiz neben dem Gesichtsmerkmal reagiert. Die sechste Frage zielt auf die Reinitialisierung ab, wenn der Proband wieder vor den Roboter tritt. Sie wurde mit dem Median vier als ausreichend bewertet.

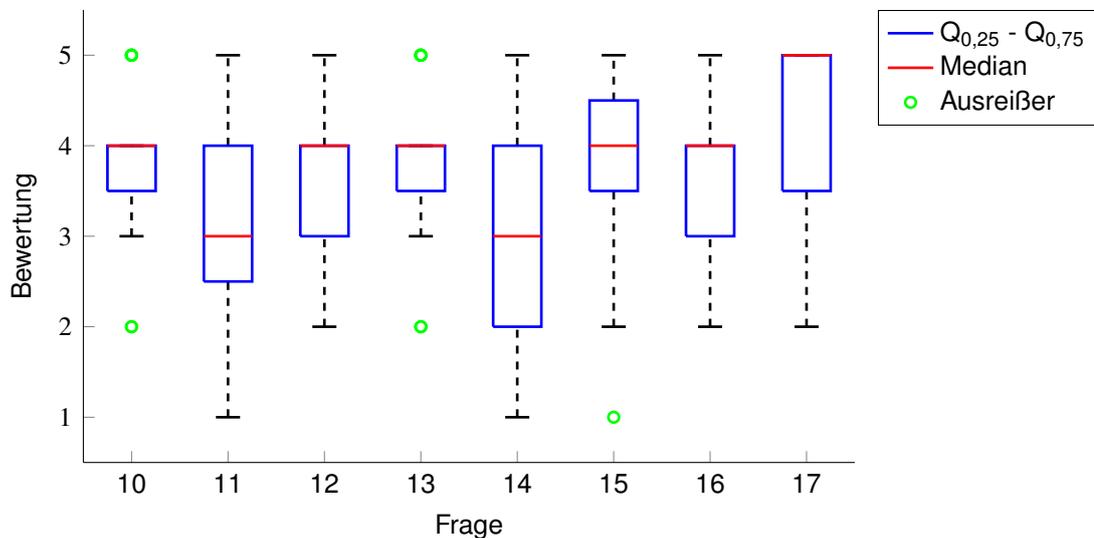


**Abbildung 5.12: Ergebnisse zur auditiven Salienz** - Diese Abbildung zeigt das Evaluationsergebnis der Probanden zur auditiven Salienz.

### Fragenblock: Auditive Salienz

- 7 - Die Reaktion des Roboters auf auditive Reize mit Dialog war realistisch.**
- 8 - Die Reaktion des Roboters auf auditive Reize ohne Dialog war realistisch.**
- 9 - Der Roboter hat mich schnell wiedererkannt, wenn ich wieder vor ihm stand.**

Wie in Abbildung 5.12 zu sehen ist, schneidet die auditive Salienz bei der subjektiven Bewertung der Teilnehmer recht gut ab. Sowohl im Dialogszenario als auch ohne Dialog liegt der Median bei vier. Dies liegt wahrscheinlich daran, dass die Reaktion auf die auditive Salienz auf einem anderen Kanal erfolgt und diese somit nur über einem Grenzwert der Umgebungslautstärke liegen muss. Auch hier wurde die Person nach den auditiven Reizen ohne Dialogsituation wieder ausreichend schnell erkannt.



**Abbildung 5.13: Ergebnisse der allgemeinen Fragen** - Diese Abbildung zeigt das Evaluierungsergebnis der Probanden auf die allgemeinen Fragen.

#### Allgemeine Fragen:

- 10 - Die Darbietung des Roboters war insgesamt erfolgreich.**
- 11 - Die Geschwindigkeit des Roboters war ausreichend.**
- 12 - Die Präzision des Roboters war ausreichend.**
- 13 - Augen- und Kopfbewegungen des Roboters wirkten harmonisch.**
- 14 - Augen- und Kopfbewegungen des Roboters sahen menschenähnlich aus.**
- 15 - Das menschenähnliche Verhalten des Roboters wirkt sich positiv aus.**
- 16 - Augen- und Kopfsteuerung könnte ich mir im realen Umfeld vorstellen.**
- 17 - Ich bevorzuge die Interaktion mit einem Roboter mit Augenbewegungen.**

Die Antworten der Probanden auf die letzten acht Fragen mit den Nummern 10 bis 17 sind in der Ergebnisgrafik 5.13 zu sehen und beziehen sich auf allgemeine Kriterien der Interaktion. Frage zehn untersucht, als wie erfolgreich die Interaktion mit dem Roboter von den Probanden bewertet wurde. Abgesehen von zwei Ausreißern wurde die Darbietung als gut bewertet. Die Geschwindigkeit des Roboters wurde als eher durchschnittlich bewertet und kann durch einen schnelleren Prozessor gesteigert

werden. Die Präzision des Roboters wurde nur als leicht überdurchschnittlich bewertet. Dies hängt wahrscheinlich mit der Genauigkeit der auditiven Lokalisierung zusammen, die nur den Winkel zur horizontalen Auslenkung bestimmt und die vertikale Verortung vernachlässigt. Die Augen- und Kopfbewegungen des Roboters wurden im Zusammenspiel insgesamt eher als harmonisch bewertet (Frage 13). Allerdings wurde in Frage 14 die Menschenähnlichkeit der Bewegung als eher durchschnittlich bewertet. Das Verhalten des Roboters wirkte sich wiederum eher positiv aus. Die Probanden konnten sich diese Steuerung im realen Umfeld gut vorstellen und bevorzugten auch die Interaktion mit einem Roboter, der über eine Augensteuerung für Dialogsituationen verfügt.

### 5.6 Zusammenfassung

In diesem Kapitel wurde die Entwicklung sowie die prototypische Umsetzung eines Blickmodells dargestellt. Dabei wurden die Komponenten zur Erfassung der visuellen sowie der auditiven Salienz und der Gesichtsmerkmale in die Roboterplattform integriert und zur Steuerung von Kopf und Augen verwendet.

Die Implementierung wurde im Rahmen eines kurzen Dialogszenarios von 20 Probanden subjektiv bewertet. Die Blicksteuerung überzeugte die Teilnehmer im Gesamteindruck und gestaltete die Interaktion mit der Plattform natürlicher und unterhaltsamer. Die Augen- sowie Kopfbewegungen konnten ebenfalls einen harmonischen Eindruck erzielen. Die Präzision des Roboters bei auditiver Salienz bietet noch Potenzial für Verbesserung, da hier lediglich in der azimutalen Ebene die Quelle der auditiven Salienz bestimmt wird.

Neben den Erkennungskomponenten des Systems kann auch das Modell weiter verbessert werden. Dazu könnten beispielsweise erkannte Emotionen [132] oder das audiovisuell bestimmte Interesse [176] des Gesprächspartners mit in das Blickmodell eingebunden werden. Wie aus der Befragung der Versuchsteilnehmer hervorging, bevorzugen die Probanden die Interaktion mit einem Roboter, der ein menschenähnliches Blickverhalten aufweist. Somit sind weitere Forschungsarbeiten auf diesem Gebiet sinnvoll und werden dazu beitragen, das Blickverhalten natürlicher und die zukünftige Zusammenarbeit von Menschen und Robotern im sozialen Umfeld noch angenehmer zu gestalten. Für weitere Arbeiten ist ein Vergleich der Blickmodelle mit einer echten menschlichen Blicksteuerung, wie beispielsweise in [177] beschrieben, interessant.

# Kapitel 6

## Lernmethoden in der Mensch-Roboter-Kooperation

Dieses Kapitel startet mit einem kurzen Überblick über verschiedene Lernmethoden im Zusammenhang mit Robotern. Auf die Beschreibung der meist gebräuchlichen Methoden folgt eine Beschreibung der im Rahmen dieser Arbeit in den Projekten JAHIR und CustomPacker verwirklichten Ansätze und Lernmethoden. Dabei wird zunächst instruktionsbasiertes Lernen für hybride Montageprozesse untersucht. Anschließend wird eine Programmierung durch Vormachen untersucht, bei der die Verpackschritte aus Punktwolkendaten extrahiert werden. Zum Schluss des Kapitels wird ein Konzept für eine intuitive Programmierung einer Verpackstation vorgestellt. Dabei werden die vorherigen Lernmethoden aufgegriffen und um eine visuelle Repräsentation der gelernten Arbeitsschritte ergänzt.

Der Begriff Lernen wird je nach Anwendungsfeld und Disziplin unterschiedlich definiert beziehungsweise verstanden. Ganz allgemein versteht man unter Lernen den Vorgang, Wissen zu erwerben oder vorhandenes Wissen zu erweitern. Dieses Wissen kann auf dem Aneignen und Merken von Fakten und Regeln basieren oder eher praktischer Natur sein und Bewegungsabläufe oder Manipulationsfähigkeiten repräsentieren.

[178] definiert Lernen als Verhaltensänderungen basierend auf Erfahrungen. Dazu gehören auch die Änderungen, die nicht auf dem Erwerb von neuem Wissen beruhen, jedoch sind von dieser Definition Verhaltensänderungen ausgeschlossen, die man nicht mit Lernen in Verbindung bringen würde.

Das Lernen aus der Sicht der Psychologie lässt sich in zwei unterschiedliche Forschungstheorien gliedern. Den Behaviorismus und den Kognitivismus [179]. Beim Behaviorismus wird der Fokus auf das Verhalten des Menschen je nach den äußerlichen

Reizen und Eingaben gelegt. Wie der Mensch die Informationen aufnimmt und verarbeitet, um zu seinem Verhalten zu kommen, interessiert hierbei nicht. Genau diese Lücke wiederum ist der Fokus des Kognitivismus.

Beim maschinellen Lernen werden üblicherweise mathematische Funktionen oder Modelle zur Lösung bestimmte Probleme gebildet. Die Parameter dieser mathematischen Konstrukte werden meist anhand von Trainingsdaten approximiert oder gelernt. Im Idealfall kann ein System nicht nur die ihm präsentierten Trainingsdaten wiedererkennen, sondern auch andere ähnliche Daten beziehungsweise Muster zuordnen [133]. Dies ist möglich, wenn das System ein allgemeineres Modell für die ihm präsentierten Daten gebildet hat. Bei maschinellem Lernen und künstlicher Intelligenz ist eine weitere Unterscheidungsmöglichkeit, in welcher Form das Wissen repräsentiert wird:

- I) Zum einen gibt es hier die symbolische Ebene, die das Wissen explizit in Form von Fakten und Regeln repräsentiert [180]. Die Verarbeitung der Fakten und Regeln erfolgt mittels Prädikatenlogik. Hierbei werden Prädikate mithilfe von Variablen zu booleschen Wahr- und Falsch-Werten evaluiert und lösen dann in Verbindung mit den hinterlegten Regeln Aktionen aus. Die Aktionen beziehungsweise das Verhalten des Systems lässt sich also aufgrund der gegebenen Fakten und Regeln gut nachvollziehen.
- II) In der subsymbolischen Ebene wird das erlernte Wissen nur implizit repräsentiert. Ein Beispiel hierfür sind neuronale Netze [180], die die Zuordnung von den Signaleingängen auf die Ausgänge lernen. Ein Betrachter kann hierbei jedoch die Entscheidung des Systems nicht nachvollziehen, sondern muss sich darauf verlassen, dass die gelernten Gewichtungen die korrekte Entscheidung liefern.

### 6.1 Überblick über Lernverfahren für Roboter

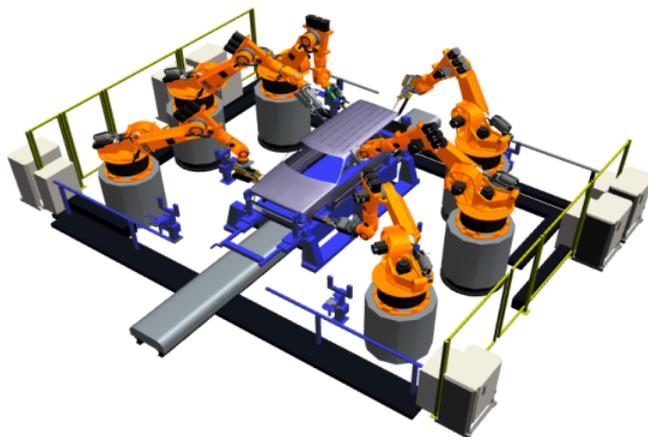
Für Roboter existieren ebenfalls unterschiedliche Lernverfahren. In dieser Arbeit wird das Lernen durch den Erwerb von neuen Regeln und Fakten untersucht und das Wissen auf symbolischer Ebene repräsentiert. Der Wissenserwerb erfolgt dabei durch unterschiedliche Demonstrationsverfahren, die in diesem Abschnitt neben weiteren Lernverfahren erläutert werden. Ein sehr interessantes Gebiet für Lernverfahren in der Industrie ist die vereinfachte Programmierung von Robotern. Hierbei helfen neue Ansätze dabei, den Roboter intuitiver zu bedienen und Arbeitsabläufe zu adaptieren.

Roboter – speziell im industriellen Umfeld – führten lange Zeit immer die exakt gleiche Aufgabe ohne größere Variationen durch. Die Produktzyklen waren wesentlich

länger als heute und somit musste die programmierte Aufgabe relativ selten angepasst oder gar komplett neu geschrieben werden. Heutzutage sind die Roboter durch Sensoren in der Lage, flexibler zu agieren und werden folglich auch für immer mehr Aufgaben eingesetzt, um in Standorten wie Deutschland dem Kostendruck durch die Globalisierung standzuhalten. Zusätzlich haben sich die Produktzyklen gerade bei elektronischen Geräten verkürzt. Inzwischen kommt als Beispiel für Mobiltelefone spätestens jedes Jahr ein neues Modell, teilweise sogar schon nach einem halben Jahr. Außerdem können die Produkte vom Kunden immer weiter individualisiert (Farbe, Gehäuse, Größe, Ausstattung, etc. ) [3] werden. Dieser Trend wird auch als kundenindividuelle Massenproduktion (engl. Mass Customization) [181] bezeichnet. Auch dieser Trend zur größeren Variantenvielfalt verlangt den Einsatz flexiblerer Roboter. Wenn die Flexibilität nicht mehr ausreicht, müssen die Programme für die Roboter angepasst oder komplett neu erstellt werden. Hierfür ist auch heutzutage oftmals noch ein Spezialist für die Roboterprogrammierung notwendig, selbst wenn die Adaption nur kleinerer Natur ist.

Hierdurch ergibt sich ein Potenzial für die Weiterentwicklung der Programmierbarkeit insbesondere für den industriellen Einsatz, um es auch Personen ohne Spezialkenntnisse zu erlauben, das Programm auf eine möglichst intuitive Art und Weise zu erstellen oder zu modifizieren. Ein möglicher Ansatz hierfür ist beispielsweise das sogenannte Einlernen (engl. Teach-In) einer neuen Trajektorie, indem der Mensch der Maschine den gewünschten Pfad einfach zeigt. Weiter entwickelte Verfahren kopieren nicht „einfach“ nur die Trajektorie, sondern versuchen, wie beim Imitationslernen in der Biologie, die demonstrierte Aufgabe zu verstehen und dadurch auch einfache Variationen dieser Aufgabe zu vollziehen [182]. Idealerweise ist das System durch das Modell und die mehrfachen Demonstrationen sogar robust gegenüber Störungen bei der Ausführung [183].

Im Folgenden soll ein Überblick über die gängigen Verfahren und Methoden zum Programmieren von Robotern gegeben werden, die einige der obigen Aspekte bereits berücksichtigen. Bei den Methoden zur Programmierung von Robotern kann man grundsätzlich zwischen Online- und Offline-Programmierung unterscheiden [184]. In dieser Arbeit liegt der Fokus aber gerade auf der Interaktion mit dem Roboter und deshalb werden Methoden zur Offline-Programmierung, wie das Simulationsprogramm KUKA Sim [185] (siehe Abbildung 6.1) oder Microsoft Robotic Studio [98], nicht weiter untersucht.



**Abbildung 6.1:** Simulationssoftware KUKA Sim [185] zur Offline-Programmierung der Roboter

### 6.1.1 Einlernen

Einlernen (engl.: Teach-In) ist eines der ersten Verfahren, um einem Roboter neue Pfade beizubringen. Bei diesem Verfahren wird der Manipulator am Roboter direkt bewegt und die Position inklusive der Rotation des sogenannten Tool Center Point (TCP) aufgezeichnet. In einigen Fällen werden auch zusätzlich auftretende Kräfte und Momente aufgenommen. Diese aufgezeichneten Informationen werden später identisch wieder abgespielt, um die Aufgabe exakt zu wiederholen.

Allerdings ist diese Variante der Programmierung prinzipbedingt durch das exakte Abspielen der Aufnahme grundsätzlich nicht robust gegenüber Störungen in der Umgebung [186, 187].

Grundsätzlich kann man beim Teach-In zwischen einer indirekten und direkten Methode unterscheiden. Bei der indirekten Methode wird der Roboterarm mittels eines geeigneten Eingabegeräts ferngesteuert an die gewünschte Position gefahren. Die Position kann dann durch Drücken einer Taste gespeichert und die nächste Position angefahren werden. Weitere benötigte Steuerlogik und zusätzliche Kommandos können noch nachträglich in der gespeicherten Datei ergänzt werden. Die Nachteile dieser Methode sind, dass sie langwierig ist, die Programmiergenauigkeit davon abhängt, wie gut der Programmierer den Bereich, in dem der Roboter agiert, einsehen kann und dass das abschließende Editieren der Datei nur wenig nutzerfreundlich ist. Trotz dieser Nachteile ist diese Methode recht weit verbreitet.

Im Unterschied zur indirekten Methode wird beim direkten Teach-In der Roboter beziehungsweise Manipulator direkt zur gewünschten Position geführt. Hierfür wird

öfters auch ein Griff zur besseren Handhabung bereitgestellt. Funktionell ist diese Methode ansonsten der indirekten sehr ähnlich. Ein Beispiel hierfür ist in Abbildung 6.2 dargestellt. Beim 'SafeTeaching - Teaching by manual guidance of a robot' der Firma MRK-Systeme GmbH [188] wird ein Kraft-Drehmomentensensor genutzt und der Roboter zusätzlich abgesichert, indem die Position, die Geschwindigkeit und die Beschleunigung überwacht werden. Wie auch bei der indirekten Methode wird hier nur der Bewegungspfad erfasst. Eventuelle Programmlogik muss zusätzlich ergänzt werden. Hierbei wird aus Sicherheitsgründen der Roboter nur bei gedrücktem Freischalttaster bewegt. Neben dem Speichern von Punkten über einen Taster besteht auch die Möglichkeit, die Punkte automatisch entweder nach einem Zeitintervall (beispielsweise 200 ms) abzutasten oder beim Überschreiten gewisser Distanzen (beispielsweise 5 mm) zu speichern.



**Abbildung 6.2:** Sicheres und direktes Teach-In der Firma MRK-Systeme GmbH [188]

Laut [189] hat die direkte Teach-In-Methode unter anderem den Vorteil, dass Programmierkenntnisse nicht unbedingt benötigt werden, Messungen des Arbeitsraums nicht zwingend erforderlich sind und die Programmierung bereits unter realen Umweltbedingungen stattfinden kann.

### 6.1.2 Lernen generalisierter Trajektorien

Eine Weiterentwicklung der Teach-In-Methode sind Verfahren, die nicht nur eine Bewegung exakt umsetzen, sondern auch mittels mehrerer Demonstrationen eine Bewegung abstrahieren können. Wie auch bei der vorherigen Methode werden hierbei mithilfe von Sensoren die Bewegungsdaten aufgezeichnet, jedoch diesmal weiterverar-

beitet und nicht direkt wieder abgespielt. Diese Verarbeitung dient dazu, das Bewegungsmuster zu verallgemeinern, um dadurch auch noch nicht im Training gesehene Bewegungen abdecken zu können. Diese Lernmethoden können zwar noch keine komplexen Aufgaben verstehen, jedoch grundlegende Bewegungsmuster nachahmen. Methoden, die auf dieser Ebene arbeiten, basieren entweder auf statistischen Methoden oder nutzen dynamische Systeme zur Umsetzung [67].

Statistische Verfahren können gut die Variationen zwischen den Demonstrationen abbilden. Eine Möglichkeit hierfür ist beispielsweise, einen generalisierten Bewegungspfad aus den Mittelwerten der aufgezeichneten Trajektorien zu generieren, der von einem durch die Varianz gegebenen Schlauch umgeben ist.

In [66] wird ein HMM verwendet, um die Daten zu komprimieren und nur die relevanten Schlüsselszenen der Bewegung abzuspeichern, die beim Training auftreten. Außerdem kann das HMM dann auch verwendet werden, um ähnliche Bewegungsmuster wieder zu generieren.

Andere Methoden berechnen eine Vielzahl von möglichen Trajektorien. Mithilfe von HMM wird die Trajektorie mit der höchsten Wahrscheinlichkeit ausgewählt. Das Mimesis-Modell folgt einem Ansatz, in dem ein HMM angewendet wird, um neue generalisierte Pfade zu berechnen. Diese Methode ist unter anderem auch in [67] erwähnt. Ein prinzipbedingter Nachteil dieser Verfahren ist die mangelnde Garantie für die Bewegungstreue. Es wird also eben nicht jedes Mal exakt die gleiche Trajektorie abgefahren, was bei industriellen Anwendungen zu Problemen führen kann.

### 6.1.3 Lernen auf semantischer Ebene

Die Verfahren auf der semantischen Ebene beschäftigen sich nicht mit der Pfadgenerierung, sondern versuchen, Hierarchien und Ablaufregeln für komplexere Aufgaben zu finden. Aus diesem Grund benötigen solche Verfahren ein System mit sehr gut ausgeprägten Fähigkeiten (engl. Skills) zum Erkennen und Greifen von Objekten, mit Pfadplanern inklusive Kollisionsvermeidung etc.

In [190] wird ein Verfahren beschrieben, das einen Roboter befähigt, Aufgaben wie das Decken eines Frühstückstischs zu lernen. Hierbei wird das Wissen über die Aufgabe mit jeder Demonstration erweitert und es werden neue Reihenfolgen sowie Möglichkeiten bei der Ausführung der Aufgabe abgespeichert.

Die Grundidee hinter diesem Konzept ist, dass die Lernmethode erkennen kann, welche bereits vorhandenen Fähigkeiten benötigt werden, um den beobachteten Vorgang zu erledigen. Das System kann dann die so erkannten Fähigkeiten selbst in der notwendigen Reihenfolge anordnen und ausführen.

Eine gängige Möglichkeit, die gelernten Sequenzen zu repräsentieren, ist, diese als Graph abzuspeichern. Hierzu wird für jede Beobachtung ein linearer Graph abgelegt, der genau die Reihenfolge der notwendigen Fähigkeiten darstellt. Ähnlich wie beim Lernen von generalisierten Trajektorien kann auch hier durch mehrerer Demonstrationen eine Varianz in den Graphen erzeugt werden. Hierzu muss das System die linearen Graphen zweier Beobachtungen, die zu einer Aufgabe gehören, miteinander kombinieren. Somit können alternative Pfade im Graphen entstehen, die es dem System ermöglichen, flexibler zu handeln und an den Knoten selbst zu entscheiden, welcher Schritt als nächstes ausgeführt wird. Somit kann das System beispielsweise Schritte auf später verschieben, die momentan nicht ausgeführt werden können, wenn der Graph dies zulässt. Wie häufig bei Systemen, die mehrere Demonstrationen für die Verallgemeinerung benötigen, muss hier eine geeignete Anzahl an Iterationen gefunden werden. Auf der einen Seite profitiert das System von einer sehr hohen Zahl an Demonstrationen, auf der anderen Seite möchte jedoch die Person, die diese Schritte durchführen muss, natürlich möglichst wenige Iterationen selbst durchführen. Nach der Trainingsphase kann der Trainer dem System bei der Ausführung des Gelernten zusehen und dabei folgende zwei Fehler beobachten:

- Das System fügt unnötige Schritte in den Graphen ein (falsche Positive)
- Das System überspringt notwendige Schritte (falsche Negative)

Hier muss der Trainer nacharbeiten und je nach Fehlerart Knoten im Graphen löschen oder in den Graphen einfügen. Möglicherweise muss er auch wieder in den Trainingsmodus wechseln und das Training mit den fehlenden Schritten wiederholen [191].

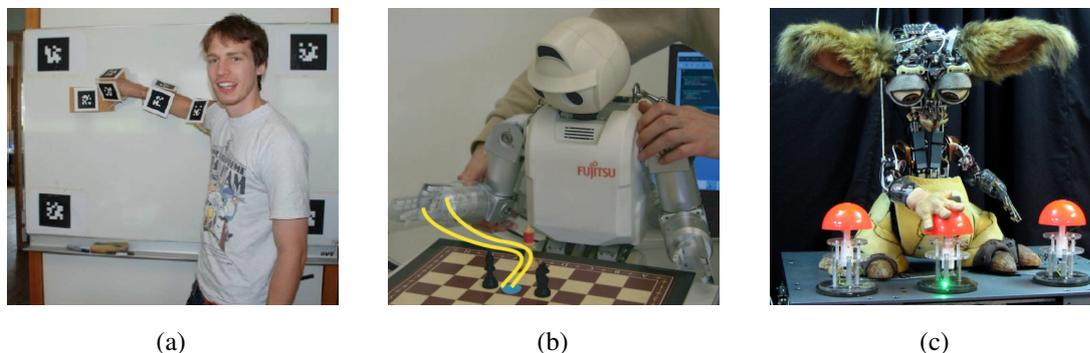
### 6.1.4 Demonstrationsverfahren

Neben den obigen Lernverfahren kann auch die Art und Weise, wie die Aufgaben beziehungsweise die Trajektorien demonstriert werden, variieren. In Abbildung 6.3 sind exemplarisch drei unterschiedliche Arten, einen Roboter zu programmieren, zu sehen.

Die Aufgabe kann vom Menschen alleine vorgemacht werden, wobei dieser nur von Sensoren beobachtet wird, um die Aufgabe oder die Trajektorie zu erlernen. Dies wird auch als Programmieren durch Vormachen (engl.: Programming by Demonstration) bezeichnet. Dieser Weg wird in Abbildung 6.3 (a) gezeigt [182] und erfordert die Transformation der extrahierten Information in die Gelenkkonfiguration des Roboters sowie eventuelle Adaptionen, die beispielsweise den Arbeitsbereich des Roboters

## 6. LERNMETHODEN IN DER MENSCH-ROBOTER-KOOPERATION

---



**Abbildung 6.3: Demonstrationsverfahren für Roboter** - (a) Programmieren durch Vormachen [182] (b) direktes Einlernen [67] (c) Lernprozess durch Instruktionen [192]

berücksichtigen. In Abbildung 6.3 (b) [67] wird die Bewegung des Roboters direkt eingelernt. Hierbei entfällt der Schritt der Transformation der Daten auf den Roboter, da die aufgezeichneten Daten bereits im Robotersystem vorliegen. Abbildung 6.3 (c) [192] zeigt den Roboter Leonardo, der unter anderem durch Instruktionen Aufgaben lernt. Diese Art wird auch als Instruktionsbasiertes Lernen (IBL) bezeichnet.

Offensichtlich sind also gewisse Demonstrationsverfahren für bestimmte Lernverfahren besser geeignet als andere. Die kinästhetische Demonstration eignet sich sehr gut für Teach-In oder das Lernen von generalisierten Trajektorien, da hier die gewünschten Informationen bereits im Robotersystem vorhanden sind. IBL eignet sich dagegen eher für den Lernprozess auf der semantischen Ebene, da das System hierbei schon wissen muss, was das bezeichnete Objekt ist und wie man dieses findet und bedient. Das Programmieren durch Vormachen kann grundsätzlich für beide Methoden verwendet werden, da hier sowohl die Trajektorie als auch die manipulierten Objekte extrahiert werden können. Neben diesen Arten der Demonstration gibt es natürlich noch weitere Ansätze. Hybride Ansätze ermöglichen dem Nutzer beispielsweise direkt den Manipulator zu bewegen sowie die zusätzliche Eingabe von Informationen per Sprache, Gesten oder Blick.

Für diese Demonstrationsverfahren gibt es eine Vielzahl von verwendbaren Sensoren (Farb- oder Tiefenbildkameras, Tracking Systeme, Taster, Joysticks, Mikrofone, Kraft-Momenten-Sensoren etc.), um die erforderliche Information für das Lernen aufzuzeichnen. Bei der Auswahl dieser Sensoren für ein spezielles Anwendungsszenario sind neben den erforderlichen Daten und deren Genauigkeit auch die Robustheit sowie die Beschaffungskosten insbesondere in der Industrie ein wichtiger Entscheidungsfaktor.

Im Folgenden werden die in dieser Arbeit untersuchten Demonstrationsverfahren zum Lernen von Prozessen weiter erläutert. Zunächst wird das IBL für hybride Montageprozesse betrachtet. Des Weiteren wurde das Lernen von Verpackungsschritten anhand von Punktwolken untersucht. Abschließend wird ein intuitives Programmierkonzept präsentiert, das für den Einsatz in einer industrienahen Verpackstation entwickelt wurde.

## 6.2 Instruktionsbasiertes Lernen für hybride Montageprozesse

Im Rahmen des Projekts JAHIR wurde IBL für die Generierung von Ablaufsteuerungen in einer hybriden Montagestation betrachtet. Die Ablaufsteuerung ist mit der multimodalen Architektur aus Kapitel 4 realisiert worden. Die wissensbasierte Kontrollinstanz steuert den Ablauf mithilfe von durch die beteiligten Komponenten bereitgestellten Fähigkeiten (engl.: Skills), wie es auch in [193] beschrieben wurde.

Dieses Lernverfahren gehört zu den Lernmethoden auf semantischer Ebene, die bereits sehr intelligente und weit entwickelte Komponenten benötigen. Dadurch können die Komponenten für die Kontrollinstanz einen hohen Abstraktionsgrad zur Verfügung stellen und müssen sich selbst um Teilaufgaben kümmern, wie beispielsweise die Pfad- und Greifplanung für die Manipulation von Objekten.

Exemplarisch für das IBL sei nun der folgende Dialog zwischen dem System und dem Nutzer gegeben:

NUTZER: Ich möchte dir eine neue Aufgabe beibringen.

SYSTEM: Ich bin bereit.

NUTZER: Warte auf Bestätigung.

SYSTEM: Schritt „Warte auf Bestätigung“ hinzugefügt.

NUTZER: Reiche rote Kiste.

SYSTEM: Schritt „Reiche rote Kiste“ hinzugefügt.

NUTZER: Wechsle in den Schraubmodus.

SYSTEM: Schritt „Wechsle in den Schraubmodus“ hinzugefügt.

NUTZER: Speichere Aufgabe als Aufgabe 4.

SYSTEM: Aufgabe gespeichert als Aufgabe 4.

Mit der ersten Anweisung versetzt der Nutzer das System aus dem Produktivmodus in den Lernmodus. Mithilfe der folgenden Regel in JESS-Notation wird die Grammatik *LG* für die möglichen Instruktionen in BNF [111] erstellt.

## 6. LERNMETHODEN IN DER MENSCH-ROBOTER-KOOPERATION

---

```
(defrule buildLearningGrammar "Extend LG"
  ?b <- (skill (speech ?sp) (appended -1))
  =>
  (bind ?*LG* (?sr combine ?*LG* ?sp))
  (modify ?b (appended 1))
)
```

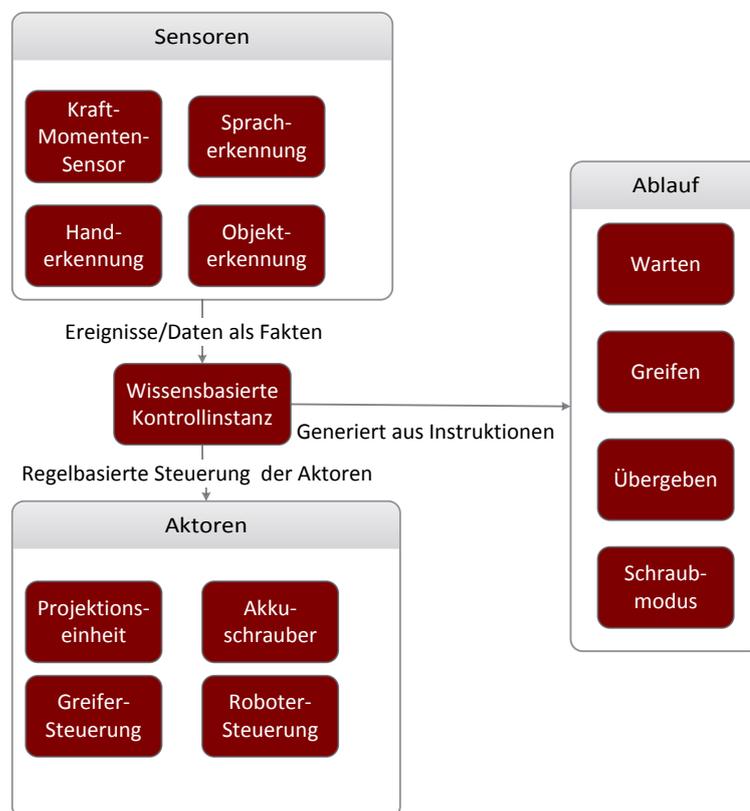
Wie in der Regel zu sehen ist, wird diese Grammatik aus den Sprachbefehlen der registrierten Fähigkeiten (engl.: skill) in der Wissensbasis erstellt. Ein Ausschnitt dieser Grammatik, reduziert auf die Fähigkeit zum Reichen von Objekten sowie ergänzt um die Instruktionen zum Rückgängigmachen, Speichern und Abbrechen, ist in folgender BNF-Darstellung gegeben:

```
<root>      ::= <baseskill> | <undo> | <save> | <cancel>
<baseskill> ::= <give> | ...
<give>     ::= Reiche <object>
<object>   ::= <color> <objecttype>
<color>    ::= rote | grüne | blaue
<objecttype> ::= Kiste
<undo>     ::= Mache letzten Schritt rückgängig
<save>     ::= Speichere Aufgabe als Aufgabe <number>
<cancel>   ::= Lernmodus abbrechen
<number>   ::= 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
...
```

Ist die Grammatik geladen und das System für weitere Instruktionen bereit, wird dies dem Nutzer per Sprachausgabe kommuniziert. Mit der Instruktion „Warte auf Bestätigung“ wird eine sensitive Interaktionsfläche (siehe Abschnitt 4.3.6) mit dem Text „Fertig“ durch das System angezeigt und der Ablauf erst nach Betätigung dieser Interaktionsfläche durch den Nutzer fortgeführt. Durch die Instruktion „Reiche rote Kiste“ wird das System angewiesen, dem Nutzer das Objekt „rote Kiste“ zu übergeben. Hierfür muss jedoch zunächst eine Komponente zum Greifen von Objekten vorhanden sein. Die Greifersteuerung muss wissen, welcher Greifer dazu verwendet wird, um diesen gegebenenfalls vorher einzuwechseln. Des Weiteren muss eine Komponente für die Objekterkennung (hier wurde die Software aus [194] verwendet) zur Verfügung stehen, die eine rote Kiste erkennen kann und die erforderlichen Fakten (Position und

Rotation im gemeinsamen Koordinatensystem) für das Greifen dieses Objekts liefert. Für den Wechsel in den Schraubmodus muss die Greifersteuerung den Akkuschauber einwechseln. Außerdem werden zur Steuerung der Schraubrichtung sensitive Interaktionsflächen eingeblendet und die Geschwindigkeit des Akkuschaubers wird anhand der Daten des Kraft-Momenten-Sensors reguliert. Abschließend wird die Aufgabe im System mit der Nummer vier abgelegt und das System wechselt wieder in den Produktivmodus.

Die Abbildung 6.4 zeigt die für die Beispielinstruktionen relevanten Sensoren sowie Aktoren und den generierten Ablauf der wissensbasierten Kontrollinstanz.



**Abbildung 6.4:** Wissensbasierte Kontrollinstanz mit Sensoren und Aktoren sowie generiertem Ablauf

Die wissensbasierte Kontrollinstanz verwendet zur Steuerung einen prädikatenlogischen Fakten- und Regelsatz. Die Fakten werden von den Sensoren aktualisiert und hinzugefügt. Die Objekterkennung kann extrahierte Objekte inklusive Typ, detektierter

## 6. LERNMETHODEN IN DER MENSCH-ROBOTER-KOOPERATION

---

Farbe, Mittelpunkt in dem geteilten Arbeitsbereich sowie Orientierung beispielsweise als Fakten in JESS-Notation in der folgenden Form ablegen:

```
(objekt (typ kiste) (farbe rote) (x 127) (y 50) (a 30))
```

Auch die anderen Sensoren legen ihre Daten in Form von Fakten ab. Diese Prädikate können mehrstellig sein – so wie im oben genannten Beispiel – aber auch einstellig. Der durch IBL gewünschte Ablaufplan ist als Regelwerk hinterlegt und steuert somit, dass die gewünschte Reihenfolge bei der Verarbeitung im Produktivmodus eingehalten wird. Wenn im Produktivmodus mehrere (beispielsweise zwei rote Kisten) oder keine Lösungen vorkommen, muss das System eine Möglichkeit haben, diesen Konflikt aufzulösen. Eine Möglichkeit besteht darin, den Nutzer zu fragen, welches der erkannten Objekte nun manipuliert werden soll. Hierzu können beispielsweise mithilfe der Projektionseinheit Ziffern eingeblendet werden, die der Nutzer dem System zur eindeutigen Identifizierung mitteilen kann. Bei jeder Aktualisierung des Arbeitswissens (der Menge an Fakten) muss überprüft werden, ob wieder eine der vorhandenen Regeln greift.

Für eine bessere Performanz ist es sinnvoll, den weiterverarbeitende Komponenten Sensordaten mit hohen Aktualisierungsraten direkt zur Verfügung zu stellen. In diesen Fällen kann die wissensbasierte Kontrollinstanz den Zugriff aktivieren beziehungsweise deaktivieren und somit auf das Verhalten der Aktoren einwirken. Ein Beispiel hierfür ist die Ansteuerung des Akkuschaubers durch die Werte des Kraft-Momenten-Sensors. Ein Anwendungsfall für dieses Lernverfahren ist in Abschnitt 7.3.1 weiter ausgeführt. Ein Videobeispiel zu der Erstellung eines Ablaufs durch Instruktionen findet sich in [195] und die Ausführung des so erstellten Ablaufs kann in [196] betrachtet werden.

Die wissensbasierte Kontrollinstanz bietet den Vorteil, den Ablaufplan in Form von Regeln zu repräsentieren und diese weitergeben zu können. Zukünftig ist es denkbar, die durch Instruktionen gelernten Abläufe und das Wissen mit anderen Robotern oder Systemen zu teilen. Ein interessanter Ansatz dazu findet sich in [197]. Dort wird ein eigenes World-Wide-Web für Roboter zum Austausch von Wissen entwickelt. Damit wäre es auch möglich, das Wissen von anderen Robotern, das beispielsweise durch Simulationen gewonnen wurde, für die Lösung eigener Aufgaben heranzuziehen. Jedoch wird noch einige Zeit vergehen, bis sich ein übergreifender Standard etabliert hat, um das Wissen einheitlich zu repräsentieren und in einen für die jeweilige Roboterplattform ausführbaren Code zu transformieren.

## 6.3 Lernen von Verpackschritten aus Punktwolken

In diesem Abschnitt wird das Programmieren durch Vormachen für Verpackprozesse untersucht. Die an einem solchen Verpackprozess beteiligten Materialien und Waren können wie in [198] definiert werden:

- *Packgut*, das die eigentlich zu verpackende Ware bezeichnet.
- *Packstoffe*, die das Material der Verpackung bezeichnen (beispielsweise Holz, Kunststoff).
- *Packmittel* bezeichnet dabei die Art der Verpackung, wie zum Beispiel Dose, Kiste, Schachtel.
- *Packhilfsmittel* sind Kleber, Klammern, Etiketten, Folien, Polster etc.

Für das hier entwickelte System wurden nur Packgut und Packmittel betrachtet, die in diesem Abschnitt mit  $O_{pg}$  beziehungsweise mit  $O_{pm}$  bezeichnet werden. Die Packhilfsmittel wurden nicht untersucht, da diese entweder schon vollautomatisiert durch spezielle Maschinen aufgebracht werden, wie zum Beispiel Kleber und Etiketten, oder aufgrund der schwierigen Handhabung (beispielsweise Folien) ohnehin rein manuell vom Menschen bearbeitet werden müssen.

Mit den extrahierten Informationen über die Art und Weise, wie Packgut und Packmittel manipuliert werden müssen, und einem geeigneten System kann später das Produkt verpackt werden. Für die Extraktion dieser Informationen wurden im Rahmen dieser Arbeit Punktwolken verwendet. Die Abbildung 6.5 zeigt die notwendigen Aktionen zur Erkennung der Verpackschritte aus der Punktwolke. Zunächst müssen in einer



**Abbildung 6.5:** Aktionen zur Verpackschritterkennung aus der Punktwolke.

Initialisierungsphase die Objekte benannt sowie Merkmale für die Erkennung gefunden werden. Diese Merkmale können entweder in einem Training erstellt werden oder aus einer Objektdatenbank geholt werden. Im Anschluss daran wird die extrahierte Punktwolke für die späteren Verarbeitungsschritte vorverarbeitet. Nach der Vorverarbeitung kann die eigentliche Erkennung der Verpackschritte beginnen. Hierfür werden

Differenzen zwischen zwei Punktwolken verwendet, um die Verschiebung der Objekte feststellen zu können. Folglich kann die erste Verpackschritterkennung erst nach der Aufnahme von zwei Punktwolken erfolgen. Eine vor der Manipulation und eine nach der Manipulation durch den Benutzer. Die extrahierte Verpackschrittinformation wird anschließend für die Imitationsphase repräsentiert. Die Aufgabe der Schlüssel-szenenerkennung ist es, die korrekten Zeitpunkte für die Aufnahme der Punktwolken für die Differenzberechnung zu erkennen. Dazu detektiert diese das Ende der Manipulation durch den Benutzer und liefert dadurch jeweils die nächste Punktwolke für den folgenden Verarbeitungsdurchlauf. Dies wird solange fortgesetzt, bis sich nur noch ein Objekt in der betrachteten Szene befindet.

Für die Verarbeitung der Punktwolke wurde PCL [90] verwendet. Im Folgenden werden die Aktionen nach der Initialisierung sowie die in den Aktionen stattfindenden Verarbeitungsschritte näher erläutert.

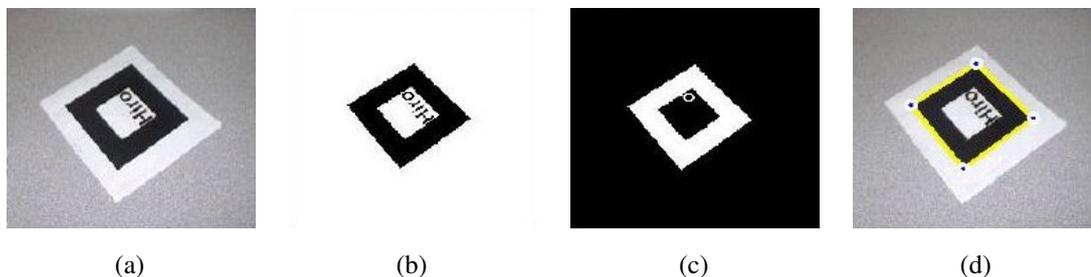
### 6.3.1 Vorverarbeitung

Bevor die eigentliche Extraktion der für den Verpackprozess relevanten Verpackschritte erfolgen kann, muss zunächst die Punktwolke in vier Schritten vorverarbeitet werden:

1. Markerdetektion mittels Augmented Reality Tool Kit (ARTK).
2. Transformation des Koordinatensystems der Punktwolke in Markerkoordinaten.
3. Reduzierung der Punktmenge in der Punktwolke.
4. Segmentierung der Tischebene.

#### 1. Markerdetektion

Für ein Referenzkoordinatensystem zwischen Roboter und Sensor wird ein Marker des ARTK verwendet. Die Punktwolken werden nach der Markerdetektion in das Markerkoordinatensystem transformiert. Zudem kann der Marker auch zur Kombination mehrerer aus verschiedenen Blickwinkel aufgezeichneten Punktwolken verwendet werden. Dafür muss natürlich der Marker in allen Ansichten detektierbar sein. Voraussetzung für diese Markerdetektion ist ein kalibriertes Kamerabild mit ermittelten intrinsischen und extrinsischen Parametern. Für die Bestimmung dieser Kameraparameter wurden



**Abbildung 6.6: Bildverarbeitungsschritte zur Markerdetektion mit ARTK [200]** - (a) Graustufenbild (b) Schwarz-Weiß-Bild (c) Segmentierung (d) Kanten und Ecken

die Methoden aus [199] verwendet. Danach kann die Position des Markers im Kamerabild detektiert werden. Hierfür wird das ARTK verwendet, das eine sehr robuste Markerdetektion bereitstellt. Die einzelnen Bildverarbeitungsschritte zur Detektion des Markers sind in Abbildung 6.6 dargestellt.

Hierzu muss das Kamerabild bereits kalibriert sein. Das in (a) dargestellte Graustufenbild der Kamera wird anhand eines beleuchtungsabhängigen Schwellwerts in schwarze und weiße Pixel binarisiert. Dieses Bild wird dann, wie in (c) zu sehen ist, segmentiert. Dabei werden durch die Betrachtung der Pixelnachbarschaft (engl.: connected components) zusammenhängende Flächen gefunden. Auf diesen wird dann die Kontur gesucht und somit werden, wie in (d) zu sehen ist, die Kanten und Ecken des Markers bestimmt. Der so bestimmte Bildausschnitt kann daraufhin extrahiert werden, um durch den Vergleich mit Vorlagen den genauen Markertyp zu bestimmen. Die ermittelten Werte der Eckpunkte des Markers in Bildkoordinaten sind somit bekannt. Dadurch kann eine Transformationsmatrix  $\mathbf{T}$  berechnet werden, die die Transformation vom Kamera-Koordinatensystem ins Marker-Koordinatensystem beschreibt.

## 2. Koordinatentransformation

Bei der Suche nach einer Transformation von einer Punktwolke  $\mathcal{P}_1$  in eine andere  $\mathcal{P}_2$  gilt für die einzelne Korrespondenz zweier Punkte  $\mathbf{p}_{1_i} \in \mathcal{P}_1$  und  $\mathbf{p}_{2_i} \in \mathcal{P}_2$  dieser Wolken die folgende Gleichung:

$$\mathbf{p}_{1_i} = \mathbf{R} \cdot \mathbf{p}_{2_i} + \mathbf{t} + \mathbf{v}_i, \quad (6.1)$$

wobei hier  $\mathbf{R}$  die gemeinsame  $3 \times 3$  Rotationsmatrix und  $\mathbf{t}$  der gemeinsame  $3 \times 1$  Translationsvektor aller Punkte sei.  $\mathbf{v}_i$  ist der Fehlerterm bei der Transformation der korrespondierenden Punktepaare.

Um diese Fehler zu minimieren und somit eine optimale Schätzung für die Transformation der beiden Punktwolken zu finden, wird üblicherweise die Minimierung der aufsummierten Fehlerquadrate (engl.: least-squares-method) verwendet. Für die Punkte ergibt sich damit die Gleichung 6.2.

$$\Sigma^2 = \sum_{i=1}^N \|\mathbf{p}_{1_i} - R \cdot \mathbf{p}_{2_i} - \mathbf{t}\|^2. \quad (6.2)$$

In Abbildung 6.7 ist das Ergebnis dieser Transformation zu sehen. Die Punktwolke wurde in das Koordinatensystem des Markers transformiert. Somit steht nun ein Referenzkoordinatensystem zur Verfügung, das auch für die Manipulation durch einen Roboter verwendet werden kann.



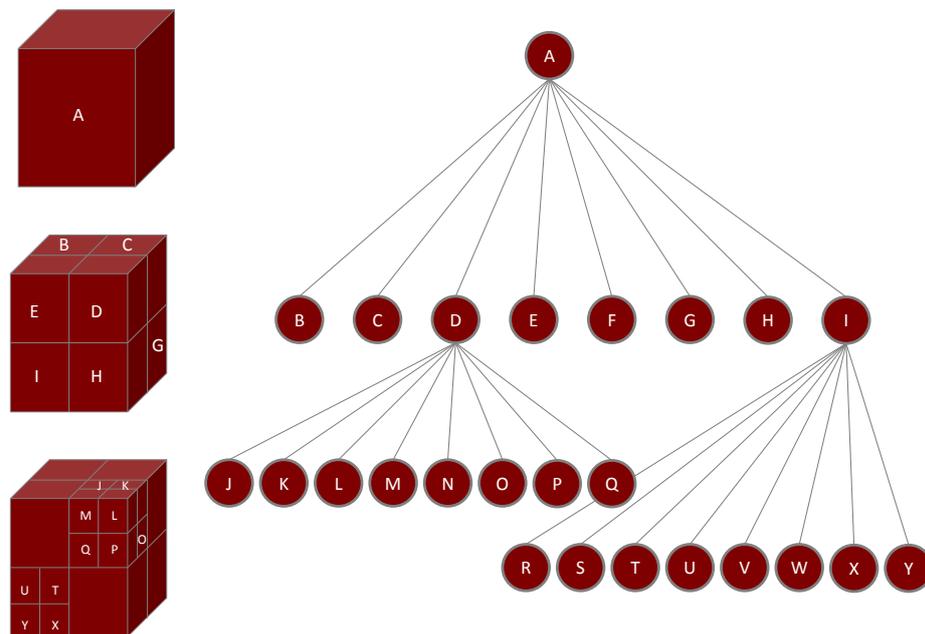
**Abbildung 6.7:** Das Koordinatensystem der Punktwolke liegt nun in der Mitte des Markers, wobei die z-Achse (blau) aus der Tischebene nach oben zeigt.

Bevor die Tischebene aus der Punktwolke segmentiert wird, muss in einem weiteren Schritt die Anzahl der Punkte reduziert werden.

### 3. Reduzierung der Punktwolke

Die Reduzierung der Punktwolke bietet den Vorteil, dass sich die Gesamtzahl der Punkte für die anschließenden Verarbeitungsschritte reduziert [60]. Außerdem werden

dabei auch durch Messrauschen erzeugte Punkte mit ähnlichen Werten zusammengefasst. Mit dieser Reduktion geht jedoch auch ein Verlust der Farb- und Tiefeninformation einzelner Punkte einher. Im Wesentlichen erfolgt die Reduzierung der dreidimensionalen Daten durch eine rekursive Einteilung des Volumens in acht Würfel [201], wie in Abbildung 6.8 dargestellt.

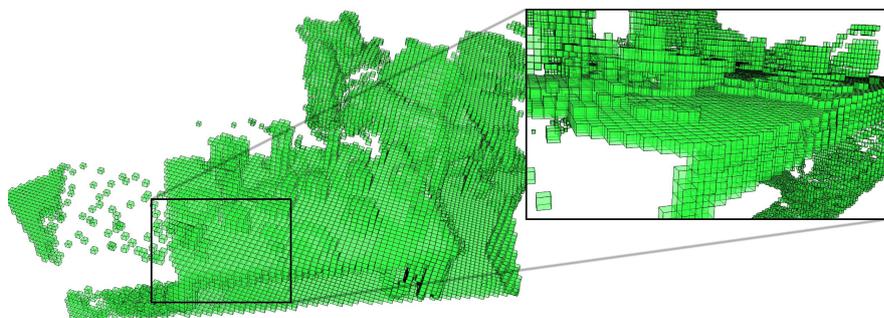


**Abbildung 6.8:** Oktalbaum: Die linke Seite zeigt die Zerlegung des Ausgangsvolumens *A* in je acht Würfel. Die Repräsentation als Oktalbaum ist auf der rechten Seite dargestellt. Vergleiche [201]

In der Abbildung ist zu sehen, dass das Ausgangsvolumen *A* in die acht Würfel *B* bis *I* unterteilt wird. Im Beispiel enthalten nur die Würfel *D* und *I* noch weitere Daten und werden daher wieder in acht Würfel unterteilt. Diese Rekursionsschritte werden solange fortgeführt, bis entweder die gewünschte Genauigkeit (Kantenlänge des Würfels) oder keine Daten mehr in einem Würfel vorliegen.

Dadurch entsteht eine hierarchische Repräsentation des Raums, die aufgrund der Einteilung in acht Würfel und der hierarchischen Anordnung als Oktalbaum (engl.: Octree) bezeichnet wird. Die Volumen enthaltenden Würfel können als sogenannte Voxel gerendert werden und repräsentieren durch den Schwerpunkt des Voxels stell-

vertretend alle darin liegenden Punkte. Ein Beispiel für eine voxelbasierte Darstellung ist in Abbildung 6.9 zu sehen.



**Abbildung 6.9:** Voxelbasierte Darstellung einer Punktwolke, die als Oktalbaum repräsentiert ist [60].

### 4. Segmentierung der Tischebene

Die Segmentierung der Tischebene erfolgt in zwei Teilschritten. Zunächst wird die Punktwolke auf den Arbeitsbereich gefiltert und daraus dann die Tischebene extrahiert.

#### I) Entfernen der Punkte außerhalb des Arbeitsbereichs

Nach der Reduzierung der Punktwolke kann diese noch auf den Arbeitsbereich gefiltert werden, um weitere nicht relevante Punkte zu entfernen. Dazu werden obere und untere Grenzen für die drei Achsen festgelegt und es gilt somit für die gefilterten Punkte der Punktwolke  $\mathcal{P}_f$ :

$$\begin{aligned}x_{min} &\leq p_{x_f} \leq x_{max}, \\y_{min} &\leq p_{y_f} \leq y_{max}, \\z_{min} &\leq p_{z_f} \leq z_{max},\end{aligned}\tag{6.3}$$

Die so gewonnene Punktwolke enthält nun idealerweise nur noch die Tischebene sowie die darauf enthaltenen Objekte. Die Abbildung 6.10 zeigt das Ergebnis dieser Verarbeitung.

Somit kann im nächsten Schritt diese Ebene segmentiert und aus der Punktwolke entfernt werden.



**Abbildung 6.10:** Punktwolke, die nur noch Punkte enthält, die in den Bereich der Tischplatte fallen. Alle anderen Punkte, die außerhalb dieses Bereichs liegen, wurden ausgefiltert.

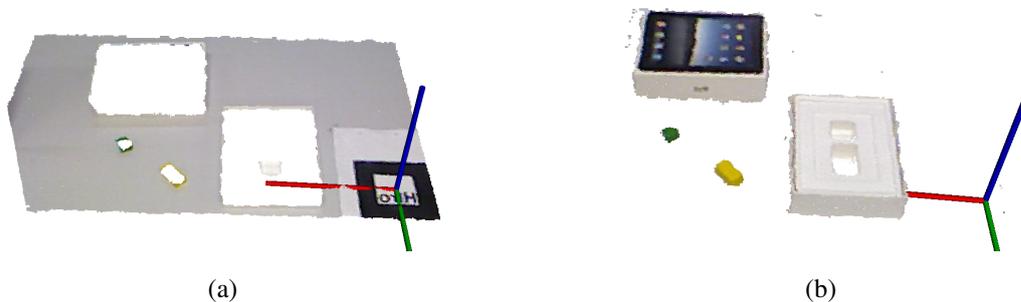
## II) Entfernen der Tischebene

Bevor die Tischebene aus der Punktwolke entfernt werden kann, muss diese erst lokalisiert werden. Hierbei macht sich der Algorithmus zunutze, dass eine Ebene bereits durch drei Punkte, die nicht kollinear<sup>1</sup> sind, beschrieben werden kann. Hierbei kommt der RANSAC-Algorithmus [69] zum Einsatz, um auch mit Ausreißern bei der Bestimmung der Ebene umgehen zu können:

1. Bestimmung von drei zufälligen Punkten  $\{\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3\}$  aus  $\mathcal{P}$  die nicht kollinear sind;
2. Bestimmung der Ebenengleichung mit den drei gewählten Punkten;
3. Distanzberechnung aller Punkte  $\mathbf{p} \in \mathcal{P}$  zu dieser Ebene;
4. Bestimmung der Anzahl der Punkte  $\mathbf{p}^* \in \mathcal{P}$ , deren Abstand  $d$  zu dieser Ebene unter einer gewählten Distanz  $d_t$  liegt und somit die Bedingung  $0 \leq |d| \leq |d_t|$  erfüllt.

Die im letzten Schritt bestimmte Punktmenge wird gespeichert. Die obigen Schritte werden nun  $k$  mal wiederholt. Der Wert für  $k$  kann mittels der Gleichung 3.10 bestimmt werden. Die Menge, die nach den  $k$  Iterationen die größte Mächtigkeit aufweist, wird

<sup>1</sup>Punkte, die nicht auf einer Geraden liegen



**Abbildung 6.11: Segmentierung der Tischebene aus  $\mathcal{P}_f$**  - (a) Tischebene mit  $d_t = 1,5$  cm. (b) Punktwolke  $\mathcal{P}_f$  ohne die in (a) dargestellte Tischebene.

als Kandidat für die gesuchte Ebene ausgewählt. Aus dieser Punktmenge wird mittels der Methode der kleinsten Quadrate die optimale Ebene bestimmt. Idealerweise sind in der Punktwolke ohne diese Ebene nur noch die für den Verpackprozess relevanten Objekte enthalten, wie es in Abbildung 6.11 (b) zu sehen ist.

### 6.3.2 Verpackschritterkennung

Dieser Abschnitt beschreibt die Methode zur Extraktion der Verpackschritte. Eine wesentliche Randbedingung ist hierbei, dass der Bereich in dem die Verpackung stattfinden muss, komplett in der Punktwolke zu sehen ist und jeweils nur ein Objekt pro Schritt manipuliert wird. Da außerdem die Pose der Objekte mittels VFH [61] bestimmt wird, müssen die Trainingssignaturen für die Objekte in der gewünschten Genauigkeit vorliegen. Diese Signaturen können vorher entweder mittels definierter Rotation des realen Objekts aufgenommen oder durch ein synthetisiertes Training mithilfe eines Computer-Aided Design (CAD)-Modell des Objekts generiert werden.

Obwohl die Punktwolke nach der Vorverarbeitung nur noch die notwendigen Punkte für die am Verpackprozess beteiligten Objekte enthalten sollte, müssen daraus noch die einzelnen Objekte detektiert werden. Dazu werden die Punkte der Wolke anhand ihres räumlichen Abstands zueinander gruppiert. Hierfür muss ein Wert für den maximalen Abstand gewählt werden, der zwischen den Punkten eines Clusters existieren darf. Die Objekte werden erkannt, indem mittels der Chi-Quadrat-Distanz (CQD) für jedes dieser Cluster die Signaturen  $S_i$  mit den Trainingssignaturen  $T_i$  verglichen werden. Damit ergibt sich die folgende Formel mit  $N = 308$  für die Anzahl der Behälter

im VFH:

$$\chi^2(S_i, T_i) = \sum_{k=1}^N \frac{(S_{i_k} - T_{i_k})^2}{T_{i_k}}, \quad (6.4)$$

Hierbei wird das Cluster dem Objekt aus dem Trainingssatz mit der geringsten Distanz zugerechnet. Das System versucht nun, die Anzahl, den Namen und den Typ der Objekte Packgut beziehungsweise Packmittel zu extrahieren. Des Weiteren wird der relative Translationsvektor des Schwerpunkts des Packmittels und des Schwerpunkts des im aktuellen Schritt verpackten Packguts sowie ein Winkel für notwendige Rotationen des Packguts relativ zum Packmittel berechnet.

Um zu bestimmen, ob ein Objekt in ein anderes verpackt wurde, wird die maximale Distanz zwischen der konvexen Hülle und dem Schwerpunkt des Objekts berechnet, wie in Algorithmus 1 beschrieben.

---

#### Algorithmus 1

Maximale Distanz zwischen der konvexen Hülle und dem Schwerpunkt eines Objektes

---

- 1: initialize  $d_{max} = 0$
  - 2: **for all**  $\mathbf{p} \in \mathcal{H}$  **do**
  - 3:   calculate euclidean distance  $d_{euk}$  between  $\mathbf{p}$  and  $\bar{\mathbf{p}}$
  - 4:   *//  $\bar{\mathbf{p}}$  is the centroid of  $\mathcal{C}$*
  - 5:   **if**  $d_{euk} > d_{max}$  **then**
  - 6:      $d_{max} = d_{euk}$
  - 7:   **end if**
  - 8: **end for**
  - 9: **return**  $d_{max}$
- 

Nach der Analyse der ersten Verpackszene wird immer ein Objekt verpackt. Die Szene direkt vor und nach einem abgeschlossenen Verpackschritt wird als Schlüsselszene bezeichnet. Die Schlüsselszenenerkennung ist in Abschnitt 6.3.4 beschrieben. Nun werden jeweils die zwei Punktwolken von aufeinanderfolgenden Schlüsselszenen verglichen. Hierbei bezeichnet  $\mathcal{P}_t$  die Punktwolke zur aktuellen Schlüsselszene und  $\mathcal{P}_{t-1}$  die Punktwolke zur vorherigen Schlüsselszene.

Aus diesen beiden Punktwolken werden nun die beiden Differenz-Punktwolken  $\mathcal{P}_{d_1}$  und  $\mathcal{P}_{d_2}$  gebildet, wie in Algorithmus 2 beschrieben wird.

Hierbei ist  $\mathcal{P}_{d_1}$  die Differenz-Punktwolke, die ermittelt wird, um das manipulierte Objekt vor dem Verpackschritt zu extrahieren.  $\mathcal{P}_{d_2}$  ist hingegen die Differenz-Punktwolke, die das Objekt nach dem Verpackschritt darstellt.

**Algorithmus 2**Bestimmung der beiden Differenz-Punktwolken  $\mathcal{P}_{d_1}$  und  $\mathcal{P}_{d_2}$ 

---

```
1: create empty point clouds  $\mathcal{P}_{d_1}$  and  $\mathcal{P}_{d_2}$ 
2: for  $i = 0$  to  $i < size(\mathcal{P}_t)$  do
3:   // size of  $\mathcal{P}_t$  equals size of  $\mathcal{P}_{t-1}$ 
4:   calculate squared euclidean distance  $d_k^2$  between  $\mathbf{p}_{t_i}$  and  $\mathbf{p}_{t-1_i}$ 
5:   if  $d_k^2 > d^2$  then
6:     //  $d^2$  is given
7:     if  $\mathbf{p}_{t_{i_z}} < \mathbf{p}_{t-1_{i_z}}$  then
8:       //  $\mathbf{p}_{*z}$ : z value of  $\mathbf{p}_*$ 
9:       add  $\mathbf{p}_{t-1_i}$  to  $\mathcal{P}_{d_1}$ 
10:    else
11:      add  $\mathbf{p}_{t_i}$  to  $\mathcal{P}_{d_2}$ 
12:    end if
13:  end if
14: end for
15: return  $\{ \mathcal{P}_{d_1}, \mathcal{P}_{d_2} \}$ 
```

---

Anhand von  $\mathcal{P}_{d_1}$  kann also festgestellt werden, welches Objekt verpackt wurde. Der Algorithmus 3 versucht zu bestimmen, in welches Objekt das manipulierte Objekt verpackt wurde. Hierzu wird nun von  $\mathcal{P}_{d_2}$  auch die konvexe Hülle  $\mathcal{H}_F$  bestimmt. Nun wird für jedes Cluster der Objekte geprüft, ob die Distanz aller Punkte der konvexen Hülle  $\mathcal{H}_F$  zum Schwerpunkt des Clusters kleiner, als die maximale Distanz ist, die für dieses Cluster mithilfe des Algorithmus 1 berechnet wurde.

Wenn dies für alle Punkte der Fall ist, gibt der Algorithmus 3 den *index* des gesuchten Clusters zurück, in das das Objekt verpackt wurde. Somit ist bekannt, welche zwei Objekte an diesem Verpackschritt beteiligt waren. Nun fehlt noch die Information über die Pose des verpackten Objekts.

Dazu wird die Transformationsmatrix von  $\mathcal{P}_{d_1}$  nach  $\mathcal{P}_{d_2}$  bestimmt. Aus der ermittelten Transformationsmatrix werden die Rotationswinkel sowie der Translationsvektor berechnet.

Da eine direkte Korrespondenz zwischen den Punkten der beiden Punktwolken  $\mathcal{P}_{d_1}$  und  $\mathcal{P}_{d_2}$  nicht bekannt ist, wird, wie in [202], Sample Consensus Initial Alignment (SAC-IA) zur Registrierung der beiden Punktwolken verwendet und damit die Transformation geschätzt. Hierfür wird eine bestimmte Anzahl an Punkten in  $\mathcal{P}_{d_1}$  gesucht,

**Algorithmus 3**Bestimmung des Objektes in das  $O_V$  verpackt wurde

---

```

1: initialize  $i = 0$ 
2: initialize  $index = -1$ 
3: for all  $\mathcal{C}$  do
4:   initialize  $k = 0$ 
5:   for all  $\mathbf{p} \in \mathcal{H}_F$  do
6:     calculate euclidean distance  $d$  between  $\mathbf{p}$  and  $\bar{\mathbf{p}}_i$ 
7:     //  $\bar{\mathbf{p}}_i$  is the centroid of  $\mathcal{C}_i$ 
8:     if  $d > d_{max_i}$  then
9:       //  $d_{max_i}$  is calculated in algorithm 1
10:      break
11:    else if  $k == \text{size}(\mathcal{H}_F) - 1$  then
12:       $index = i$ 
13:      return  $index$ 
14:    end if
15:     $k = k + 1$ 
16:  end for
17:   $i = i + 1$ 
18: end for
19: return  $index$ 

```

---

die alle einen Abstand größer als  $d_{min}$  zueinander haben müssen. Danach werden für diese Punkte aus  $\mathcal{P}_{d_1}$  Punkte mit einem ähnlichen Histogramm aus  $\mathcal{P}_{d_2}$  gesucht. Mittels dieser Korrespondenzen wird wiederum eine Transformation und durch eine Fehlermetrik die Qualität dieser Transformation berechnet. Für die Fehlermetrik wurde auch die Huber Fehlerfunktion [203] verwendet, da hier der Einfluss von Ausreißern nicht so stark wie beispielsweise beim rein quadratischen Fehlermaß ist. Nach einer bestimmten Anzahl an Wiederholungen des Algorithmus wird die Transformation mit der besten Qualität basierend auf der Fehlermetrik ausgewählt.

Die dadurch bestimmte Transformation wird jedoch nicht direkt gespeichert, da diese Transformation speziell für die observierte Orientierung und Lage der Objekte in dieser Szene gilt. Daher werden nur die sogenannten Offsetwinkel  $\langle \alpha_{O_i}, \beta_{O_i}, \gamma_{O_i} \rangle$  bestimmt. Dazu wird die Rotation in die einzelnen Rotationen um die x-, y- und z-Achse aufgeteilt, wie es in Gleichung 6.5 zu sehen ist.

$$R = R_x(\alpha_F) \cdot R_y(\beta_F) \cdot R_z(\gamma_F), \quad (6.5)$$

Die Aufteilung erfolgte durch eine QR-Zerlegung mittels Givens-Rotation. Die daraus bestimmten Rotationswinkel können in die gesuchten Offsetwinkel durch Einsetzen in die Gleichung 6.6 ermittelt werden.

$$\begin{aligned} \alpha_O &= \alpha_F - (\alpha(O_{pg}) - \alpha(O_{pm})), \\ \beta_O &= \beta_F - (\beta(O_{pg}) - \beta(O_{pm})), \\ \gamma_O &= \gamma_F - (\gamma(O_{pg}) - \gamma(O_{pm})), \end{aligned} \quad (6.6)$$

Diese drei Winkel geben die relative Rotation des zu verpackenden Packguts  $O_{pg}$  zum Packmittel  $O_{pm}$  in das es verpackt wird, an. Genau wie diese relativen Winkel wird nun auch die relative Position des zu verpackenden Objekts  $O_{pg}$  zum Packmittel  $O_{pm}$  benötigt. Dazu wird der Translationsvektor der beiden Schwerpunkte der Objekte berechnet und mit der inversen Rotationsmatrix von eventuellen Rotationen bereinigt. Die so aus der Demonstrationsphase extrahierten Informationen können nun repräsentiert werden, um in der Imitationsphase den Verpackprozess zu synthetisieren.

### 6.3.3 Repräsentation

Die so extrahierten Informationen zu den Objekten und den erkannten Verpackschritten müssen nun noch repräsentiert werden. Diese Informationen können für die weitere Verarbeitung in Form von Fakten und Regeln in Präfixnotation abgelegt werden.

Für Objekte wie das Packgut „Stein g“ (gelber Stein) sieht eine mögliche Darstellung folgendermaßen aus:

```
(objekt
(typ "pg")
(name "Stein g")
(status "uvp")
(position x y z alpha beta gamma)
)
```

Passend zu diesen Objekten werden Regeln zum Verpackprozess in der Wissensbasis angelegt, die die gelernte Information aus der Demonstrationsphase repräsentieren. So eine Regel lautet zum Beispiel, wenn es ein Objekt vom Typ „pg“ mit dem Namen „Stein g“ gibt und dieses noch den Status „uvp“ (unverpackt) hat und es ein Objekt vom

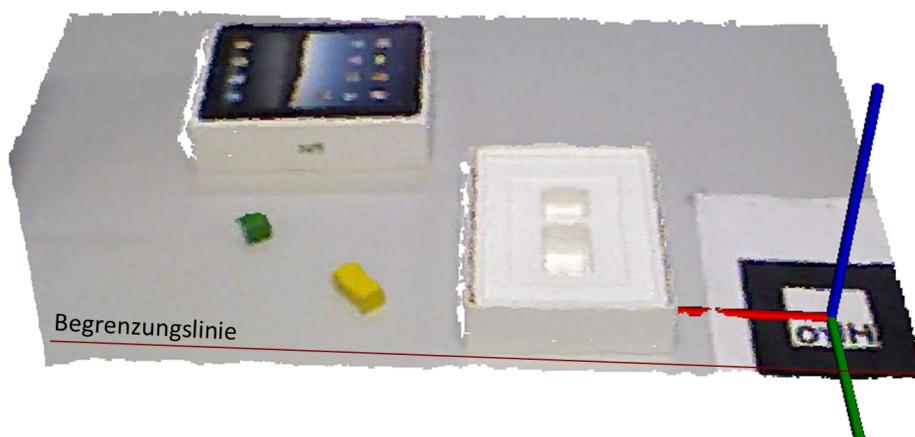
Typ „pm“ mit dem Namen „ipad boden“ gibt, dann verwende die Fähigkeit „packe“ auf dem Roboter und verpacke das Packgut in das Packmittel mit dem gespeicherten Translationsvektor und dem gespeicherten Offsetwinkeln und ändere danach den Status des Packguts auf „vp“ (verpackt).

```
(defrule step-1
?idpg <- (objekt (typ "pg") (name "Stein g") (status "uvg"))
?idpm <- (objekt (typ "pm") (name "ipad boden"))
=>
(skill (agent ?robot) (command "packe") (args ?idpg ?idpm 0
0 0 0.005, 0.009, -0.023))
(modify ?idpg (status "vp"))
)
```

### 6.3.4 Schlüsselszenenerkennung

Die letzte Aktion in der Verarbeitungskette zur Erkennung der Verpackschritte ist die Schlüsselszenenerkennung. Diese Aktion dient dazu, den Zeitschritt für die Aufnahme der nächsten Punktwolke der Szene zur Bildung der Differenz-Punktwolke zu bestimmen. Im Wesentlichen müssen sich hierfür die Hände des Benutzers außerhalb der Szene befinden, da ansonsten unfertige Manipulationsschritte verarbeitet werden würden. Um die Segmentierung zuverlässig zu gestalten, wurde eine Begrenzungslinie, wie in Abbildung 6.12 zu sehen ist, am vorderen Ende der Bearbeitungsebene überwacht. Die Linie verläuft parallel zur roten x-Achse des Marker-Koordinatensystems. Wenn der Mensch die Objekte auf der Ebene manipuliert, weichen die z-Werte (blaue Achse) für die Punkte auf dieser Linie von den Referenzwerten der Initialisierung ( $z \approx 0$ ) ab. Erst wenn der Mensch die Hände wieder aus dem Arbeitsbereich nimmt, sind hier wieder die Referenzwerte gegeben und eine Schlüsselszene kann gespeichert werden. Dies setzt voraus, dass der Mensch jeweils nur ein Objekt verpackt und dann die Hände wieder komplett aus dem überwachten Bereich nimmt. Außerdem darf bei der Manipulation kein Objekt auf die Begrenzungslinie gebracht werden.

Eine alternative Möglichkeit zur Erkennung der Szene ist (ähnlich wie bei 2D-Bilddaten die Summierung der Differenzbildpixel) eine kontinuierliche Analyse der Unterschiede der einen Punktwolke zur nächsten. Bei sehr geringen Unterschieden (die beispielsweise durch Rauschen der Messwerte entstehen können) liegt wahrscheinlich keine Manipulation durch den Benutzer vor. Trotzdem kann bei dieser Methode nicht ausgeschlossen werden, dass beispielsweise durch kurzes Verweilen des Benutzers an



**Abbildung 6.12:** Begrenzungslinie zur Schlüsselszenenerkennung

einer Stelle die berechnete Energie unter den Schwellwert sinkt und somit fälschlicherweise eine Schlüsselszene aufgenommen wird. Außerdem ist die Methode rechenintensiv, da im Prinzip für alle Werte die Differenz zu den vorherigen gebildet werden muss.

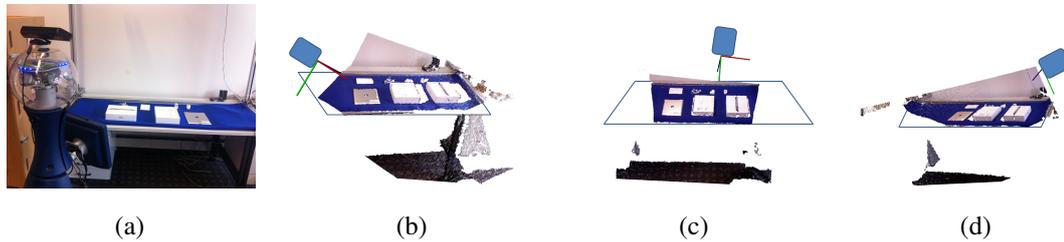
### 6.3.5 Auswertung

Die Erkennung der Objekte und die Bestimmung ihrer Posen hängt stark von deren Auflösung ab. Exemplarisch ist hierfür in Tabelle 6.1 die Erkennungsrate für Objekte mit weniger als 1000 Punkten und mehr als 1000 Punkten in der Punktwolke gegeben.

	< 1000 Punkte	> 1000 Punkte
Pose	35 %	92 %
Objekt	56 %	98 %

**Tabelle 6.1:** Erkennungsergebnisse im Zusammenhang mit der Anzahl der Punkte für die Objekte

Die Objekte wurden je 50 mal mit unterschiedlicher Pose an verschiedenen Positionen aufgenommen. Daraufhin wurde einerseits die Erkennung des Objekts evaluiert und andererseits die Schätzung der Pose. Wie zu sehen ist, erfolgt die Wiedererkennung der Objekte zuverlässiger als die Detektion der Objektpose. Des Weiteren werden größere beziehungsweise besser aufgelöste Objekte mit mehr als 1000 Punkten mit 98 %



**Abbildung 6.13: Sensorposen auf der Roboterplattform** - (a) Setup mit Roboter ELIAS  
(b) linke Perspektive (c) mittlere Perspektive (d) rechte Perspektive

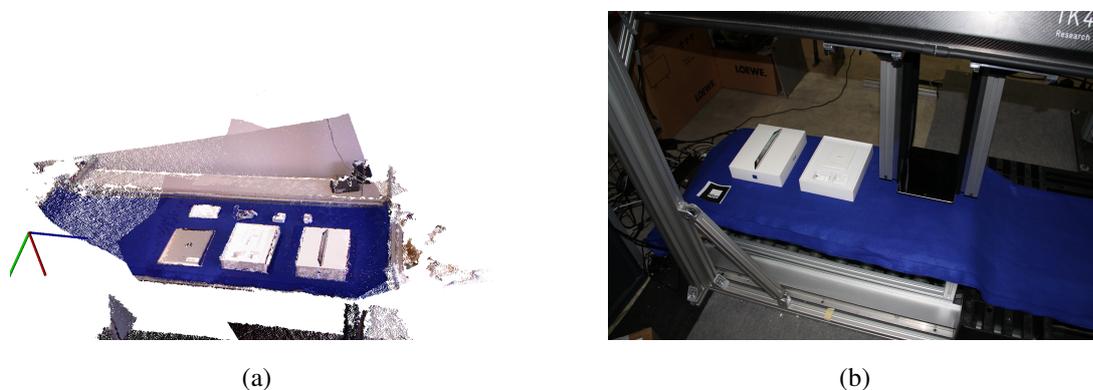
im Mittel sehr robust erkannt sowie deren Pose mit 92 % im Mittelwert ebenfalls sehr gut detektiert. Für den Einsatz des entwickelten Systems ist es von Vorteil, Kleinteile (Schrauben, Muttern, Stecker etc.) in größeren Kisten zusammenzufassen, die sich besser detektieren lassen.

Für eine bessere Auflösung der Szene wurden mehrere Punktwolken mittels eines Kinect-Sensors auf der mobilen Roboterplattform ELIAS (siehe Abschnitt 7.1.1) erzeugt. In Abbildung 6.13 (a) ist der Aufbau im Labor samt Roboter zu sehen. Des Weiteren sind die Punktwolken von den drei Aufnahmepunkten links, mittig und rechts vom Tisch dargestellt. Da der Marker des ARTK nicht in allen Perspektiven zu sehen ist, werden diese Punktwolken anhand von SAC-IA [202] zu einer Punktwolke zusammengeführt. Anschließend wird mittels des Iterative Closest Point (ICP)-Algorithmus [204] die Ausrichtung der Punktwolken zueinander optimiert. Ein erfolgreiches Ergebnis dieser Registrierung der Punktwolken ist in Abbildung 6.14 (a) dargestellt.

Bevor die extrahierten Verpackungsschritte für eine Imitation in der realen Anwendung verwendet werden können, müssen die Objekte mit Greifpunkten versehen werden. Dies ist nötig, da die extrahierten und gespeicherten Informationen nur die Objekte sowie deren ideale Transformation ineinander betreffen. In der realen Anwendung muss die Greifplanung geeignete Greifpunkte für die Manipulation am Objekt und die Pfadplanung des Roboters einen Pfad ohne Kollision zur Ablage der Objekte an den gewünschten Positionen finden.

In der Anwendung im Projekt CustomPacker wurde für die Greifplanung eine auf reconstructMe [205] basierende Software und eine Pfadplanung des Projektpartners Profactor GmbH verwendet. Abbildung 6.14 (b) zeigt, wie der Roboter im CustomPacker-Demonstrator das iPad handhaben kann. Aufgrund der Größe des Manipulators war lediglich das Greifen des iPads sowie des Deckels möglich.

Des Weiteren kann auch eine Verdeckung des Packguts durch das Packmittel und somit eine veränderte Punktwolke ein Problem für die korrekte Erkennung der Aus-



**Abbildung 6.14: iPad-Verpackprozess** - (a) zusammgeführte Punktwolken (b) Greifen des iPads mit dem CustomPacker Roboter

richtung darstellen. Das Problem der Verdeckung kann möglicherweise durch die Verwendung mehrerer Sensoren gelöst werden. Eine weitere Einschränkung des Systems ist, dass in jedem Schritt nur ein Objekt verpackt werden darf. Auch das Verpacken von Objekten in einem Packmittel, das dann selber wieder verpackt wird, wurde nicht weiter untersucht.

Grundsätzlich können die Objekte durch die Verwendung der Punktwolkendaten gut segmentiert werden. Aufgrund der Tiefendaten funktioniert dies auch gut, wenn sie eine dem Hintergrund ähnliche Farbe aufweisen. Eine weitere Verbesserung des Systems wäre beispielsweise durch die zusätzliche Verwendung der Texturinformation der Objekte zu erreichen. Diese Farbinformation könnte sich sowohl positiv auf die Klassifizierung der Objekte als auch auf die Posenerkennung auswirken. Zusätzlich könnte die Texturinformation auch bei fehlender Tiefeninformation helfen, die Informationen zu berechnen. Zu fehlenden Tiefeninformationen kann es durch reflektierende Oberflächen oder andere Infrarotlichtquellen wie beispielsweise Sonnenlicht kommen. Ein Beispiel zum Verlust der Tiefeninformation bei reflektierenden Oberflächen ist in Abbildung 6.15 dargestellt.

Die Abbildung zeigt auf der linken Seite das Farbbild der Objekte (ein iPad, ein Mac-Mini, sowie ein Spiegel) sowie auf der rechten Seite das Tiefenbild. In den schwarzen Bereichen fehlt die Tiefeninformation. Je nachdem, wie die Objekte zur Kamera liegen, kann es hier zu Reflektionen kommen, die eine Berechnung der Tiefeninformation an diesen Stellen verhindern. Diese fehlende Tiefeninformation kann durch Interpolation von den Rändern berechnet werden, wie dies in [206] für Ebenen, Zylinder und Kugeln beschrieben wird. Außerdem kann das Fehlen der Information auch



**Abbildung 6.15:** Spiegelungen bei Objekten (iPad, Mac-Mini, Spiegel): Links RGB-Bild, rechts Tiefenbild

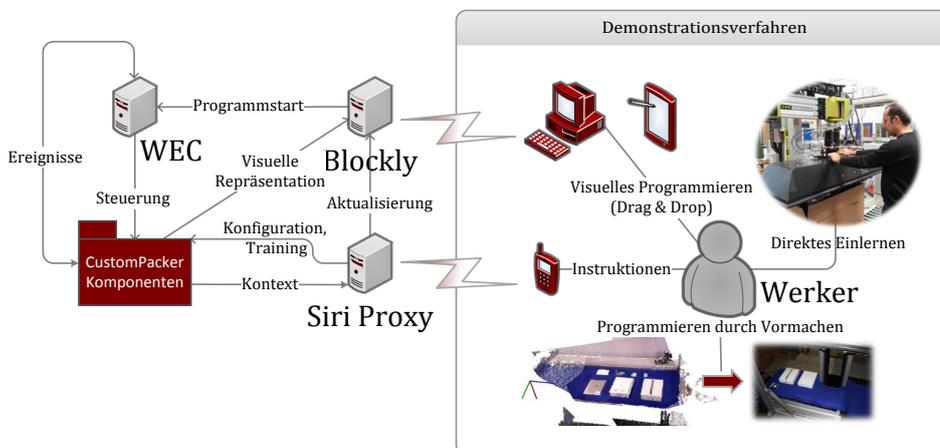
direkt als Objekt (mit bekannter Geometrie) interpretiert werden. Dazu müssen jedoch andere Quellen als Ursache für das Fehlen der Information an dieser Stelle sicher ausgeschlossen werden können.

Im nächsten Abschnitt werden das Programmieren durch Instruktion sowie das Programmieren durch Vormachen mit einer grafischen Repräsentation des Arbeitsablaufs kombiniert. Durch die Verwendung von natürlicher Sprache für die Instruktionen sowie eine leichte Modifikation des Arbeitsablaufs durch die grafische Repräsentation auf mobilen Geräten wird die Programmierung recht intuitiv.

## 6.4 Intuitive Programmierung einer Verpackstation

Dieser Abschnitt präsentiert ein Konzept zur Programmierung einer Verpackstation, das ebenfalls im Rahmen des Projekts CustomPacker entwickelt wurde. Für das entwickelte Konzept wurde von den Komponenten (Manipulationssteuerung, Assistenzsystem, Werkerpositionserkennung etc.) des Projekts CustomPacker ausgegangen, die Schnittstellen für die Konfiguration und das Training zur Verfügung stellen. Die Konzeptidee ist in Abbildung 6.16 grafisch dargestellt.

Der Werker hat dabei die Möglichkeit, das passende Demonstrationsverfahren selbst zu wählen, um den Ablauf in der Zelle festzulegen. Greif- und Roboterpositionen können durch direktes Einlernen gespeichert werden. Die Verpackschritte des Verpackprozesses können, wie im vorherigen Verfahren, durch Vormachen programmiert werden. Außerdem können auch Gesten zur Interaktion mit dem System vorgemacht werden. Durch Instruktionen kann das System auch per Sprache bedient werden, wobei in diesem Fall eine natürlichsprachliche Eingabe verwendet wurde. Zusätzlich zu den vorherigen Verfahren, bei denen der Ablauf lediglich im Expertensystem als Regelmenge

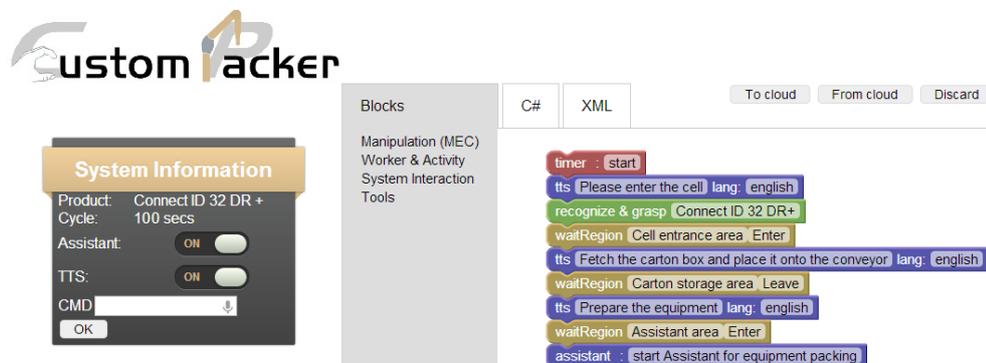


**Abbildung 6.16:** Konzept für die intuitive Programmierung der Verpackstation

hinterlegt war, wird in diesem Konzept eine visuelle Repräsentation der Schritte integriert. Für die Umsetzung der visuellen Repräsentation des Ablaufs wurde Blockly [207] gewählt. Dafür müssen die Komponenten eine visuelle Repräsentation der Fähigkeiten in Form von Blöcken bereitstellen. Der Werker kann mit diesen Blöcken den Arbeitsablauf einfach durch Klicken, Ziehen und Loslassen erstellen sowie nachträglich die Blöcke verschieben, um den Arbeitsablauf anzupassen.

Des Weiteren können diese Blöcke durch den Anwender modifiziert werden. Dies kann der Nutzer über Instruktionen, direktes Einlernen oder auch Editieren in der visuellen Darstellung starten. Die Blöcke werden vom System automatisch über die bereitgestellten Schnittstellen konfiguriert und bei Bedarf direkt als neuer Block in die visuelle Programmieroberfläche eingefügt. Die visuelle Darstellung eines Ablaufs ist beispielsweise im rechten Teil der Abbildung 6.17 zu sehen und kann sowohl vom Anwender schnell erfasst als auch vom Roboter sowie den angebenen Komponenten verarbeitet werden. Der durch die Verschaltung der Blöcke generierte Maschinencode (C#) wird durch den Programmstart zur Arbeitsablaufsteuerung (engl.: Workflow Execution Control (WEC)) an den Zentralrechner durch WebSockets übermittelt.

Die GUI wurde mit Hyper Text Markup Language (HTML) 5-Technologie realisiert. Dadurch gelingt eine einheitliche und geräteübergreifende Darstellung sowohl auf PCs als auch auf mobilen Geräten in Browsern. Die Browser sind durch WebSockets mit der WEC verbunden, wodurch eine schnelle bidirektionale Kommunikation ermöglicht wird. Dies ist nötig, damit der Browser Änderungen am Arbeitsablauf über Blockly an die WEC senden und umgekehrt die WEC dem Nutzer Aktualisierungen bezüglich des momentanen Systemzustands visualisieren kann. Ein wesentlicher



**Abbildung 6.17:** GUI-Konzept im CustomPacker. Visuelle Programmieroberfläche, basierend auf Blockly [207], mit den Blöcken für den Arbeitsablauf.

Teil der GUI ist die visuelle Repräsentation des Arbeitsablaufs. Die so dargestellte Ablaufstruktur kann hier schnell erfasst und einfach neu angeordnet werden. Im Produktivmodus wird diese visuelle Repräsentation dazu verwendet, den aktuellen Schritt im Ablauf zu markieren und den Nutzer über weitere für den Arbeitsablauf relevante Daten zu informieren. Diese Daten betreffen beispielsweise die aktuelle Produktvariante oder die Zykluszeit. Des Weiteren hat der Nutzer einige Konfigurationsmöglichkeiten, wie beispielsweise die Sprachausgabe oder die visuelle Assistenzfunktion zu deaktivieren. Dies ist auch im linken Teil der Abbildung 6.17 dargestellt.

Die WEC steuert die Komponenten an und reagiert wiederum auf Ereignisse der Komponenten. In diesem Konzept kann der Benutzer eine GUI, Spracheingabe und Sprachausgabe sowie Gesten zur Interaktion verwenden. Die Benutzerschnittstelle in diesem System verhält sich unterschiedlich je nachdem, welcher Modus des Systems gerade aktiv ist. Das System kann sich entweder im Lernmodus oder im Produktivmodus befinden. Im Lernmodus ist die Benutzerschnittstelle dafür verantwortlich, dem Benutzer die Programmierung des Arbeitsablaufs zu ermöglichen. Für die Erkennung der natürlichsprachlichen Instruktionen des Nutzers wurde Siri von Apple über einen Proxy-Server [208] verwendet. Die Integration auf mobilen Endgeräten erfolgte durch eine entwickelte Web-Applikation, die in Abbildung 6.18 zu sehen ist.

Nach der Auswahl der CustomPacker-WEC-Applikation auf dem Startbildschirm in Abbildung 6.18 (a) (dritte Reihe, ganz rechts) kann die Spracherkennung Siri von Apple aufgerufen werden, wie in (b) dargestellt. Nach der natürlichsprachlichen Eingabe wird die Rückmeldung über die Ausführung der Instruktion sowohl als Text dargestellt als auch per Sprachausgabe wiedergegeben, wie in Teil (c) zu sehen ist. Die verwendete Sprachausgabe [118] dient als Rückmeldung an den Nutzer über die er-

## 6. LERNMETHODEN IN DER MENSCH-ROBOTER-KOOPERATION



**Abbildung 6.18: CustomPacker-WEC-Applikation** - Die Schritte für die sprachbasierte Programmierung: (a) Startbildschirm mit CustomPacker-WEC-Applikation (b) CustomPacker-WEC-Applikation mit Siri (c) Erkannte Eingabe mit Systemrückmeldung

folgreiche Verarbeitung, zur Meldung von Fehlern oder zur Nachfrage bei kritischen Instruktionen. Im Produktivmodus wird die Sprachsynthese hingegen primär als Kanal für Anweisungen des Assistenzsystems zum Arbeitsablauf genutzt.

Aus der natürlichsprachlichen Instruktion wird mithilfe von regulären Ausdrücken die wesentliche Information extrahiert. Nachfolgend wird ein Beispiel für den zugehörigen regulären Ausdruck für die Instruktion aus der Abbildung 6.18 (c) angeführt:

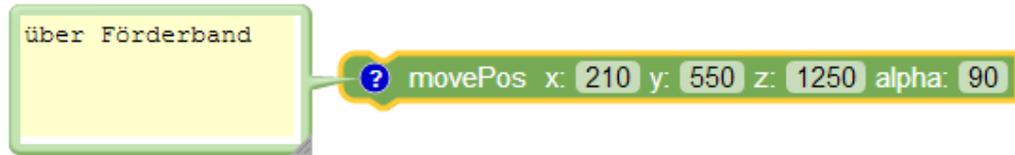
```
/.*position.*kommentar(.*)/
```

Dieser reguläre Ausdruck extrahiert aus der Instruktion zur Speicherung der Position den Kommentar für das zugehörige Feld in der visuellen Repräsentation. Dadurch wird der Block aus Abbildung 6.19 für die visuelle Repräsentation im Arbeitsablauf erzeugt.

Dafür wird neben dem Kommentar die aktuelle Position des Roboters sowie die Rotation des Manipulators erfasst und für die weitere Verarbeitung im Block hinterlegt. Neben den Instruktionen, die direkt in der Erzeugung einer visuellen Repräsentation münden, ist es auch möglich, Kommandos zur direkten Ausführung zu geben.

Hierfür kann der Roboter beispielsweise mithilfe des folgenden regulären Ausdrucks relativ zur aktuellen Position verfahren werden.

```
/.*roboter.* ([0-9,]*[0-9]) (cm|mm|meter)
```



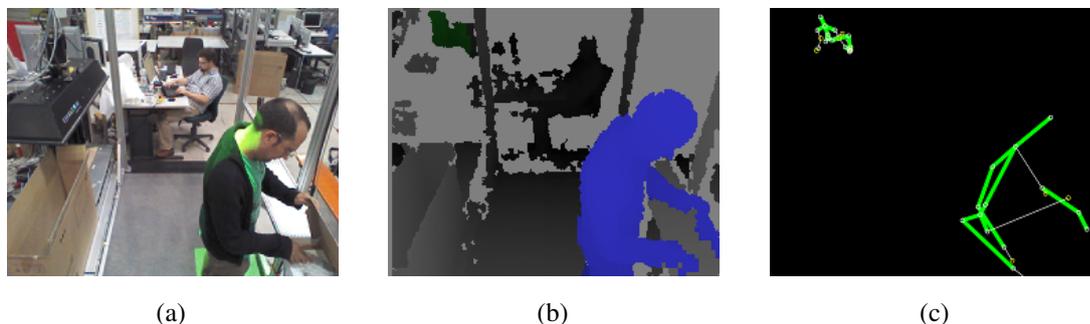
**Abbildung 6.19:** Erzeugter Block aus natürlichsprachlicher Instruktion für die visuelle Repräsentation des Arbeitsablaufs

nach (oben|unten|links|rechts|vorne|hinten) /

Aus Sicherheitsgründen erfolgt eine Rückfrage des Systems, ob der Befehl korrekt verstanden wurde. Neben der aktuellen Position des Roboters muss zur Ausführung des Befehls die Größeneinheit angegeben und vom System umgerechnet werden. Damit die Richtungsangaben (links, rechts, vorne, hinten) richtig interpretiert werden können, muss zusätzlich die Position des Werkers, der den Befehl gibt, mitberücksichtigt werden.

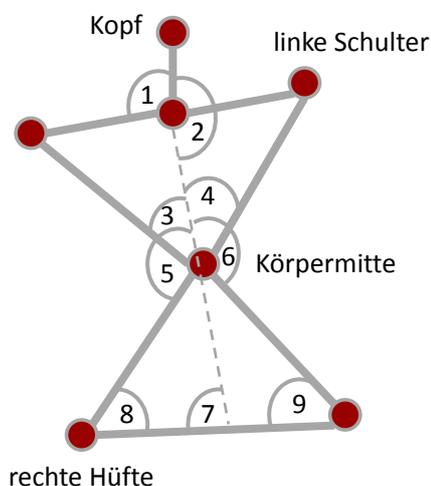
Die Werkerpositionserkennung kann neben der Kontextinformation zur Auflösung der Richtungsangaben bei natürlichsprachlichen Instruktionen auch für eine verbesserte Synchronisierung des Arbeitsablaufs zwischen Mensch und Roboter verwendet werden. Mittels der vom Kinect-Sensor gelieferten und der nachfolgend beschriebenen Verifikation dieser Personenhypothese kann diese Werkerposition in der Zelle erkannt werden. Im Lernmodus können dadurch für den Arbeitsablauf besonders relevante Bereiche gelernt werden. Hierfür muss sich der Nutzer lediglich in den gewünschten Bereich begeben und über Sprache oder das mobile Gerät das Lernen eines neuen Bereichs starten. Dann wird über ein Zeitfenster von  $t_f = 2$  Sekunden die Position der Körpermitte  $\mathbf{p}_{km}$  (siehe Abbildung 6.21) aufgezeichnet und anschließend der Mittelwert des Positionsvektors gebildet und zusammen mit dem vom Benutzer vergebenen Bezeichner des Bereichs abgelegt. Im Produktivmodus kann die Komponente das Betreten oder Verlassen des zuvor programmierten Bereichs detektieren und als Ereignis an die WEC zur Ablaufsynchronisierung weiterleiten.

Sowohl bei bewegten Objekten in einer vom Kinect-Sensor beobachteten Szene als auch bei direkter Bewegung des Sensors kommt es zu falsch detektierten Personenhypothesen. In Abbildung 6.20 (c) ist beispielsweise die Person im Vordergrund korrekt detektiert, die sitzende Person wurde nicht detektiert und durch den bewegten Roboter wurde oben links eine fehlerhafte Hypothese einer Person eingefügt. Das Einfügen der fehlerhaften Hypothese liegt daran, dass der Sensor eigentlich mit einer Hintergrundsegmentierung arbeitet und davon ausgeht, dass sich nur Personen vor dem Sensor



**Abbildung 6.20: Personenhypthesen** - (a) RGB-Kamerabild (b) Tiefenbild mit farbigen markierten Personenhypthesen (c) Skelettstruktur für die Personenhypthesen

bewegen. Während diese Hintergrundsegmentierung im sehr statischen Wohnzimmerbereich sehr gut funktioniert, scheitert das Verfahren, wenn sich neben den Personen auch noch andere Objekte bewegen. Aus diesem Grund wurde die gelieferte Personenhypthese mittels einer gewichteten Summe über Winkel in der Skelettstruktur aus Abbildung 6.21 verifiziert.



**Abbildung 6.21: Skelettstruktur mit Winkeln**

Die dargestellte Struktur zeigt den Rumpf der Personenhypthese. Des Weiteren sind die Winkel zwischen den verschiedenen Körperteilen (Kopf, Nacken, linke und rechte Schulter, Körpermitte, linke und rechte Hüfte) angedeutet und von oben nach unten durchnummeriert. Für die Verifikation der Personenhypthese wird nun mittels folgender gewichteter Summe über die Winkel ein Wert  $\sigma$  berechnet:

$$\sigma = \sum_{i=1}^{i=9} w_i b_i \tag{6.7}$$

Dabei stellt  $w_i$  ein empirisch ermitteltes Gewicht für den Winkel dar und  $b_i$  einen Binärwert, der angibt, ob der Winkel innerhalb eines Toleranzbereichs liegt oder nicht. Die Tabelle 6.2 zeigt eine Übersicht über die Winkel  $i$ , deren Gewicht  $w_i$  und die jeweiligen plausiblen Winkelbereiche als Grundlage für die Binärwertberechnung von  $b_i$ .

Winkel-Index $i$	Gewicht $w_i$	Winkelbereich
1	0,1	$90^\circ \pm 20^\circ$
2	0,2	$180^\circ \pm 20^\circ$
3	0,1	$45^\circ \pm 15^\circ$
4	0,1	$45^\circ \pm 15^\circ$
5	0,1	$135^\circ \pm 15^\circ$
6	0,1	$135^\circ \pm 15^\circ$
7	0,2	$90^\circ \pm 20^\circ$
8	0,05	$90^\circ \pm 10^\circ$
9	0,05	$90^\circ \pm 10^\circ$

**Tabelle 6.2:** Gewicht und Bereiche der Winkel in der Skelettstruktur

Bei Experimenten mit  $N = 45$  Personenhypothesen von denen 38 falsch und sieben korrekt waren, wurde das beste Ergebnis für eine Annahme der Hypothese für  $\sigma \geq 0,4$  für die gewichtete Summe aus Gleichung 6.7 ermittelt. Die sich daraus ergebende Konfusionsmatrix ist in Tabelle 6.3 dargestellt.

	$P$	$\bar{P}$
$V$	100 %	2,7 %
$\bar{V}$	0 %	97,3 %

**Tabelle 6.3:** Konfusionsmatrix für Personenverifikation mit Schwellwert 0,4

Dabei steht  $P$  für die korrekte und  $\bar{P}$  für eine falsche Personenhypothese. Äquivalent steht  $V$  für die Verifikation der Personenhypothese, also  $\sum_{i=1}^{i=9} w_i b_i \geq 0,4$  und  $\bar{V}$  für die Ablehnung der Hypothese durch den Verifizierer.

Der Verifizierer hat somit alle echten Personenhypothesen verifiziert. Allerdings wurden mit diesen Einstellungen auch 2,7 % der falschen Hypothesen irrtümlich als korrekt verifiziert. Zusammenfassend liefert der Verifikator ein brauchbares Ergebnis, um falsche Hypothesen zu verwerfen. Mit weiterer Zusatzinformation, wie beispielsweise möglichen Aufenthaltsbereichen für Personen und der Position von weiteren beweglichen Objekten (beispielsweise anderer Roboter), kann die Reduzierung falscher Personenhypothesen noch weiter verbessert werden.

### 6.5 Zusammenfassung

In diesem Kapitel wurden verschiedene Lernmethoden für die Mensch-Roboter-Kooperation betrachtet. Das instruktionsbasierte Lernen für hybride Montageprozesse wurde in einer Anwendung erfolgreich prototypisch umgesetzt. Die im Wortschatz reduzierte Spracherkennung funktionierte in der industriellen Umgebung zwar sehr robust, jedoch war der Nutzer dadurch bei der Bedienung und der Vergabe von Bezeichnern für Objekte und Aufgaben eingeschränkt.

Die Erkennung von Verpackungsschritten aus Punktwolkendaten lieferte bei den verwendeten Produkten recht gute Ergebnisse. Es bestehen hierbei jedoch noch Herausforderungen bei der Erkennung und Lokalisierung reflektierender Objekte. Außerdem müssen für die Imitation Greifpunkte für die bezüglich der Handhabung empfindlichen elektronischen Teile hinterlegt werden. Des Weiteren ist es notwendig, kollisionsfreie Pfade für die Manipulation zu finden.

Das Konzept zur intuitiven Programmierung kombiniert mehrere Verfahren und ergänzt diese um eine visuelle Repräsentation der erlernten Arbeitsabläufe. Das entwickelte Konzept wurde in einer Anwendung für das Verpacken von Fernsehern prototypisch in einer Verpackstation realisiert. Außerdem wurde die prototypische Realisierung im Rahmen einer Nutzerstudie evaluiert. Die Beschreibung der Anwendung sowie die Ergebnisse der Nutzerstudie finden sich im folgenden Kapitel in Abschnitt 7.3.2.

# Kapitel 7

## Anwendungen

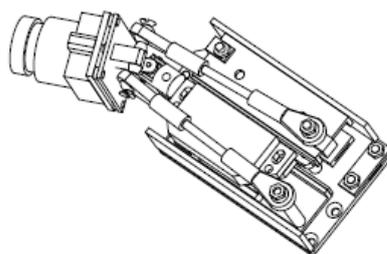
In diesem Kapitel werden Anwendungen vorgestellt, die auf Basis der vorangegangenen Kapitel umgesetzt wurden. Zu Beginn werden die dafür verwendeten Roboterplattformen und Forschungsplattformen vorgestellt. Im Anschluss daran folgen die Anwendungen im sozialen Umfeld, mit dem Fokus auf der multimodalen Interaktion zwischen Mensch und Roboter. Daran schließen sich die Anwendungen im industriellen Umfeld an, wobei auch eine multimodale Kooperation in einer Fertigungszelle betrachtet wird. Den Schluss bildet eine Anwendung zur intuitiven Programmierung in der CustomPacker-Forschungsplattform inklusive einer Nutzerstudie.

### 7.1 Roboter- und Forschungsplattformen

In diesem Abschnitt werden die für die Anwendungen im sozialen und industriellen Umfeld verwendeten Roboter- und Forschungsplattformen vorgestellt. Zunächst werden die beiden Roboterplattformen ELIAS und EDDIE eingeführt. Die Anwendungen dieser beiden Plattformen sind im sozialen Umfeld angesiedelt, wobei mithilfe von ELIAS ein Spieleszenario umgesetzt wird und EDDIE eine Getränkebestellung in einer Küchenumgebung aufnimmt. Für das industrielle Umfeld in der kognitiven Fabrik wird zunächst die Forschungsplattform des Projekts JAHIR für hybride Montageprozesse vorgestellt. Im Anschluss daran wird die im Rahmen des Projekts CustomPacker entwickelte Forschungsplattform zur Verpackung elektronischer Konsumgüter mit Unterstützung durch den Roboter präsentiert.



(a)



(b)

**Abbildung 7.1: Roboterplattform ELIAS:** - (a) Interaktionseinheit mit Kopf und berührungsempfindlichem Monitor (b) Aufbau des Roboterauges aus [210]

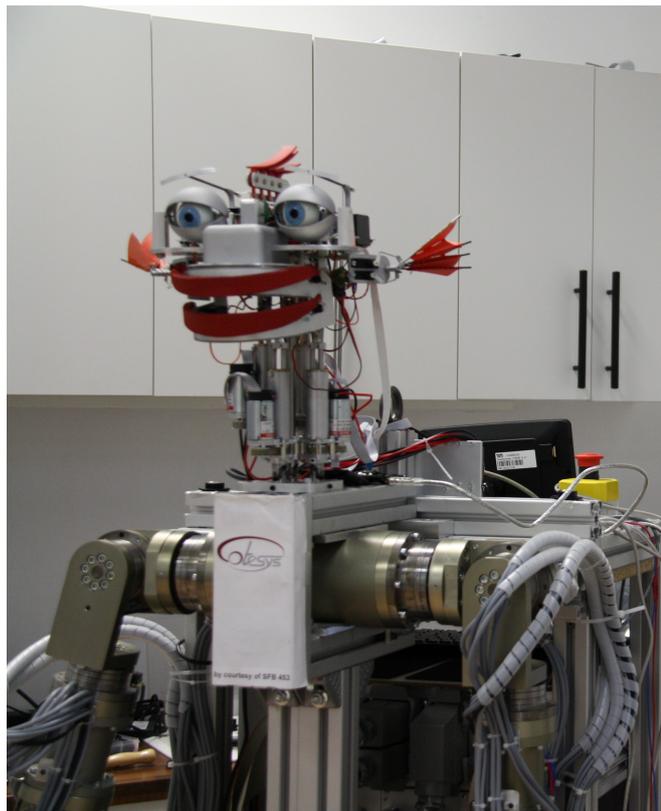
### 7.1.1 Roboterplattform ELIAS

Die Roboterplattform ELIAS wurde unter anderem für die Implementierung und Evaluierung der Blicksteuerung aus Kapitel 5 eingesetzt. Die Plattform ist in Abbildung 7.1 (a) zu sehen und stellt eine Modifikation eines Scitos G5 der Firma MetraLabs GmbH [209] dar. Der Antrieb des Roboters basiert auf einem differentiellen Antriebsprinzip und verfügt über zwei angetriebene Räder sowie ein Stützrad zur Stabilisation. Der Roboter wurde von Beginn an für den Einsatz zusammen mit Menschen entworfen und wurde dafür auch vom TÜV zertifiziert. Für den Fall, dass es zu einer Berührung mit dem Menschen kommt, detektiert eine Sicherheitsleiste in Bodennähe den Kontakt und stoppt den Roboter umgehend.

Diese Grundausstattung wurde zusätzlich um einen auch in der horizontalen Ebene drehbaren Tiefenbildsensor auf dem Kopf der Roboterplattform erweitert sowie um extrem schnell bewegbare Augen. Die Augen werden mittels eines Piezo-Antriebs basierend auf [211] bewegt. Die mechanischen Augen [210], wie in Abbildung 7.1 (b) zu sehen, übertreffen die beim menschlichen Auge maximal auftretende Rotationsgeschwindigkeit von  $700 \frac{\circ}{s}$  und -beschleunigung von bis zu  $25.000 \frac{\circ}{s^2}$  [212] um das Fünffache. Zudem wurde ein Mac-Mini an die mobile Stromversorgung der Plattform angeschlossen, um zusätzliche Rechenkapazität für die Prozesse der Mensch-Maschine-Interaktion bereitzustellen.

Der Roboterkopf verfügt über 32 LEDs, deren Intensität angesteuert werden kann. Zusätzlich kann der Kopf aus der horizontalen Ebene um bis zu  $20^\circ$  nach oben und bis zu  $7^\circ$  nach unten bewegt werden. Des Weiteren hat der Kopf in der horizontalen Ebene einen Bewegungsspielraum von  $350^\circ$ . Für den Tiefenbildsensor wurde eine Möglichkeit zur Bewegung in der horizontalen Ebene entwickelt, um den Sensor nachführen zu können. Hierzu wurde eine konisch zulaufende Platte aus Aluminium mit einer Aussparung für einen Servomotor in Zusammenarbeit mit dem Institut für Werkzeugmaschinen und Betriebswissenschaften (*iwb*) angefertigt, damit der Sensor mit dem Kopf in Richtung der Interaktion ausgerichtet werden kann. Dadurch wird verhindert, dass die gesamte Plattform rotiert werden muss.

### 7.1.2 Roboterplattform EDDIE



**Abbildung 7.2:** Mobile Roboterplattform des LSR im CCRL mit Kopf EDDIE [119]

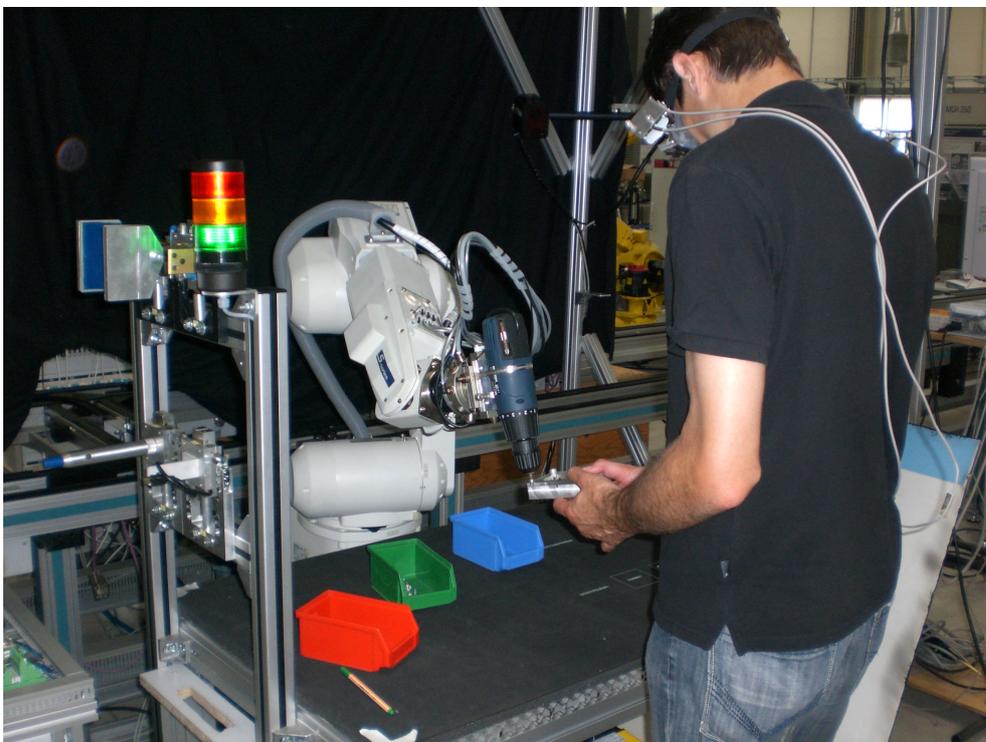
Die Abbildung 7.2 zeigt eine der omnidirektionalen mobilen Roboterplattformen vom LSR, die im CCRL eingesetzt wurden. Auf dieser Roboterplattform wurde die

## 7. ANWENDUNGEN

---

Architektur zur multimodalen Interaktion aus Kapitel 4 für Studien und Anwendungen zur Interaktion im sozialen Umfeld angewendet. Der Roboter verfügt über einen Kopf zur Darstellung von Emotionen (EDDIE [119]). Eine Kamera mit Mikrofon zur Aufnahme von Videobildern sowie zur Erfassung der Spracheingaben des Interaktionspartners kann unterhalb des Kopfes montiert werden. An den beiden Armen mit je sieben Freiheitsgraden sind Zwei-Finger-Greifer sowie Kraft-Momenten-Sensoren montiert. Zur Kollisionsvermeidung verfügt die Plattform über Laserscanner an den Ecken. Falls es doch zur Kollision kommt, ist an dem Roboter eine Sicherheitsleiste zur Notabschaltung am unteren Ende der Plattform montiert. Auf der mobilen Roboterplattform ist ausreichend Platz für die Montage mehrerer Rechner vorhanden. Dadurch können eigene Rechner für die Steuerung der Plattform, anspruchsvolle Bildverarbeitungsaufgaben sowie ein eigener Rechner für die Steuerung der Mensch-Roboter-Interaktion verwendet werden.

### 7.1.3 JAHIR



**Abbildung 7.3:** Werker bei der Schraubmontage in der hybriden Fertigungszelle JAHIR

Der Roboter im Projekt JAHIR ist ein industrieller Knickarmroboter mit sechs Freiheitsgraden (Mitsubishi RV-6SL), der fest in eine hybride Fertigungszelle integriert ist. Der Roboter verfügt sowohl über einen Kraft-Momenten-Sensor als auch über ein Greiferwechselsystem, wodurch der Roboter benötigte Werkzeuge (Klebepistole, Akuschrauber, Backengreifer, Palettengreifer) an seinem TCP fixieren kann. In Abbildung 7.3 ist ein Werker bei einer Schraubmontage zu sehen, wofür der Roboter zur Assistenz automatisch den Akuschrauber als Werkzeug selektiert hat.

Über dem Arbeitsbereich sind Farb- und Tiefenbildkameras sowie eine Projektionseinheit montiert. Des Weiteren wurden ein kopfgebundenes Mikrofon zur Spracherkennung sowie die Blickerfassungsbrille EyeSeeCam [125] verwendet. Außerdem verfügt die Forschungsplattform über eine Bandanbindung, die zum An- und Abtransport von Teilen auf Produktpaletten zwischen den anderen Montagestationen (automatische sowie rein manuelle Fertigung) innerhalb der kognitiven Fabrik [213] dient. In JAHIR wurde die multimodale Interaktion mit der Architektur aus Kapitel 4 realisiert. Des Weiteren kam auf dieser Plattform auch das instruktionsbasierte Lernen aus Abschnitt 6.2 zur Anwendung.

### 7.1.4 CustomPacker



**Abbildung 7.4:** Werker im Produktivmodus in der CustomPacker Prototyp-Zelle

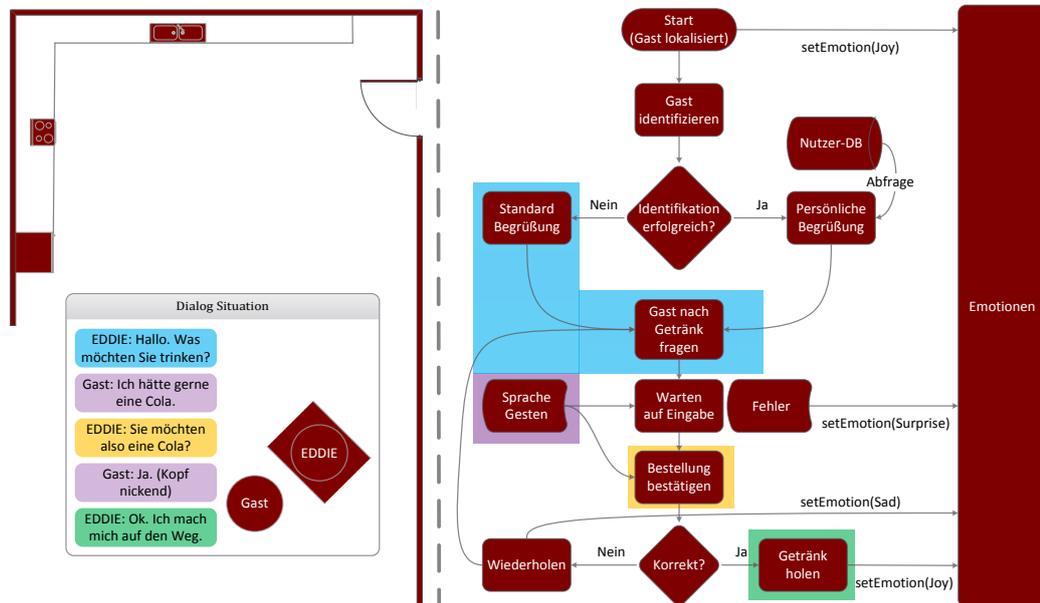
Für die Forschungsplattform CustomPacker wurde am Lehrstuhl *iwb* in der Fabrikhalle eine Prototyp-Zelle aufgebaut, die in Abbildung 7.4 zu sehen ist. Obwohl es bereits erste kommerzielle nachgiebige Roboter gibt, wie beispielsweise den UR10 [214] von Universal Robots (Traglast 10 kg) oder den KR5 SI [215] (Traglast 5 kg) der Firma MRK-Systeme (zertifiziert für den Betrieb ohne trennende Schutzeinrichtungen), wurde im Projekt ein neuer nachgiebiger Roboter entwickelt, um die erforderlichen Lasten von bis zu 30 kg manipulieren zu können. Der verwendete Roboter ist eine Neuentwicklung von FerRobotics [216] und kann Lasten bis zu 50 kg handhaben. Die Steuerung des Roboters ist von KEBA [217] und erfolgte mithilfe von Nachrichten an Funktionsblöcke, realisiert gemäß dem ereignisbasierten Automatisierungsstandard IEC 61499 [218]. Für die sichere Zusammenarbeit mit dem Menschen verfügt der Roboter über Sensoren in den drei Gelenken, die den Roboter in der horizontalen Ebene bewegen können. Des Weiteren verfügt der Roboter über einen aktiven Kontaktflansch, der die Last am TCP grundsätzlich in der Schwebelage hält. Somit können Kräfte, die nach oben oder unten einwirken, gemessen werden. Diese können entweder dazu benutzt werden, den Roboter auch in der Vertikalen zu verfahren, oder um unerwünschte Krafteinwirkungen, beispielsweise bei einer Kollision, zu detektieren. Diese Forschungsplattform diente unter anderem zur Durchführung der Nutzerstudie zur intuitiven Programmierung aus Abschnitt 6.4.

## 7.2 Soziales Umfeld

### 7.2.1 Getränkebestellung

Eine Anwendung im sozialen Umfeld, die mittels der multimodalen Architektur und der Interaktionskanäle aus Kapitel 4 realisiert wurde, ist eine Getränkebestellung. Das Demonstrationsszenario wurde in der Küche einer Laborumgebung realisiert. Diese Umgebung ist schematisch auf der linken Seite von Abbildung 7.5 dargestellt.

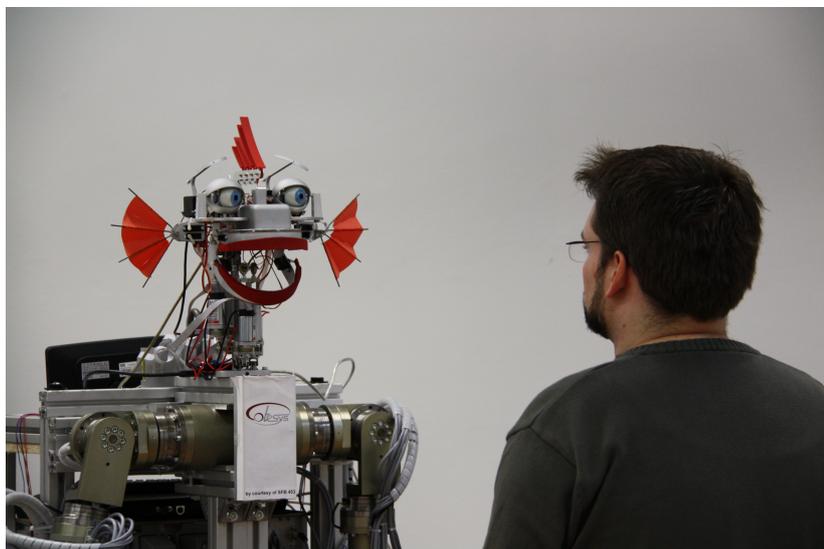
Zunächst wird der Gast mittels mehrerer an einem Gitter an der Decke der Laborumgebung befestigter Kameras lokalisiert und verfolgt [220]. Wurde der Gast erfolgreich lokalisiert, nähert sich der Roboter dem Gast. Sobald der Roboter in der Nähe der Person ist, initiiert die wissensbasierte Kontrollinstanz den Dialogfluss. Dieser ist auf der rechten Seite der Abbildung 7.5 visualisiert. Die für den Dialogschritt zuständigen Komponenten sind farbig hinterlegt. Die wissensbasierte Kontrollinstanz versucht, das Prädikat  $orderDialog(P,D)$  zu erfüllen, wobei die Variable  $P$  für die Person steht und



**Abbildung 7.5:** Links: Aufbau der Laborumgebung im CCRL [219]; rechts: Dialogfluss für die Getränkebestellung

$D$  für das bestellte Getränk. Das Prädikat wird dann anhand von definierten Äquivalenzregeln in die Prädikate  $isIdentified(P)$  &  $isOrdered(D)$  &  $isConfirmed(D)$  zerlegt. Dadurch wird die Aufgabe immer weiter in kleinere Unteraufgaben gegliedert, bis die Prädikate schließlich im Aufruf einzelner Fähigkeiten enden, die die Variablen mit den Werten (beispielsweise erkannte Person) belegen.

Zu Beginn werden Kopf und Augen der Plattform auf den Gast ausgerichtet, um dem Gast die Aufmerksamkeit des Systems zu kommunizieren und es wird versucht, den Gast mittels Gesichtserkennung zu identifizieren. Dafür kommt das System von [64] zum Einsatz, das auf der zuvor detektierten Gesichtshypothese [221] aufbaut. Wenn die Gesichtserkennung erfolgreich war, wird eine personalisierte Begrüßung abgerufen. In diesem Fall kann auch auf bereits erfolgte Bestellungen zurückgegriffen und direkt nach dem am häufigsten bestellten Getränk gefragt werden. Anderenfalls hat der Gast die Möglichkeit, die Bestellung per Spracheingabe aufzugeben. Um dem Gast die Bereitschaft des Systems zur Spracheingabe zu visualisieren, wurden die Ohren des Kopfes aufgefächert. Das System versichert sich durch eine Rückfrage, ob die Bestellung richtig verstanden wurde. Hier kann der Nutzer wiederum entweder per Sprache oder durch Kopfgesten seine Zustimmung oder Ablehnung ausdrücken.



**Abbildung 7.6:** EDDIE und Gast im Dialog

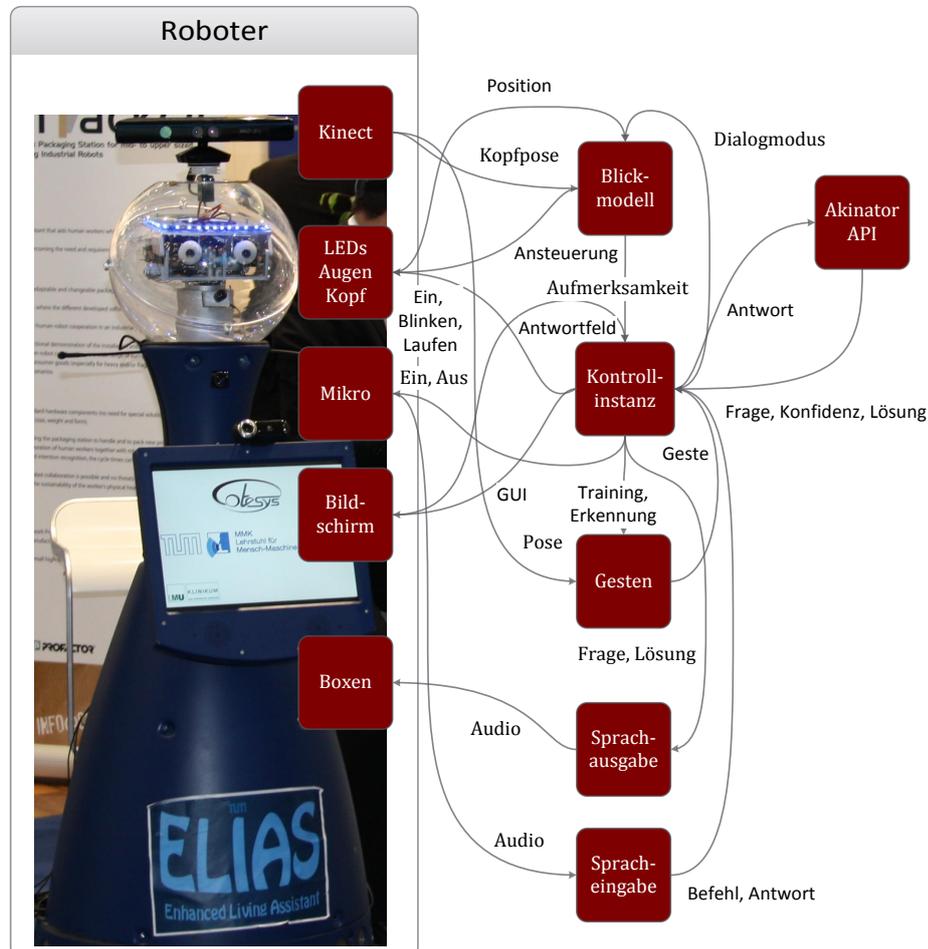
Zusätzlich werden während des Dialogs die Emotionen Freude, Überraschung und Traurigkeit mithilfe des Roboterkopfes dargestellt. Der Dialogfluss endet mit dem erfolgreichen Aufnehmen der Bestellung und der Übergabe an die Robotersteuerung, die das bestellte Getränk organisiert.

### 7.2.2 Spiel Akinator

Bei dem Spiel Akinator [222] muss der Nutzer an eine Person denken, und das System versucht, die Person mithilfe von Fragen zu erraten, indem es anhand der Antworten durch einen Entscheidungsbaum zu einem Blatt navigiert und dadurch die gesuchte Person ermittelt. Der Benutzer hat dabei immer fünf gleiche Antwortmöglichkeiten: *Ja, Nein, Ich weiß nicht, Wahrscheinlich, Wahrscheinlich nicht.*

Für die Implementierung auf der Roboterplattform ELIAS sind die Komponenten sowie der Informationsfluss zwischen diesen Komponenten in Abbildung 7.7 dargestellt:

Die wissensbasierte Kontrollinstanz koordiniert den Ablauf des Gesamtsystems. Ist der Nutzer vom System detektiert worden, hat er zu Beginn die Möglichkeit, dem System neue Gesten [174] für die fünf Antwortmöglichkeiten einzulernen. Hierfür wird der Nutzer gebeten, für jede Antwortmöglichkeit die entsprechende Geste vorzuführen. Die jeweilige Aufnahme wird durch den Wechsel der LEDs in den laufenden Modus signalisiert.



**Abbildung 7.7:** Komponenten und Informationsfluss auf der Plattform ELIAS

Im Anschluss daran startet das eigentliche Spiel. Zur Implementierung des Spiels auf der Roboterplattform wurde die bereitgestellte Akinator-API zur Wissensdatenbank verwendet, um die Fragen zu beziehen. Die Frage wird per Sprachausgabe vorgelesen und zusammen mit den Antwortmöglichkeiten auf dem Bildschirm dargestellt. Der Nutzer kann nun entweder ein Antwortfeld auf dem berührungssensitiven Bildschirm selektieren, die Antwort durch Spracheingabe mitteilen oder eine der zuvor trainierten Gesten ausführen. Die Antwort des Nutzers wird über die Akinator-API zurückgemeldet. Nun wird die Konfidenz des Systems über die gesuchte Person überprüft. Bei einer Konfidenz über die Lösung von mehr als 90 % wird die Lösung abgerufen und ausgegeben. Ansonsten wird die nächste Frage abgerufen und wiederum

ausgegeben. Dies wird so lange wiederholt, bis die Konfidenz über die Lösung den Schwellwert erreicht hat.

Die Kopfpose, die bereits für das Blickmodell aus Kapitel 5 verwendet wird, fand in diesem Szenario zusätzliche Verwendung. Die Ausrichtung des Kopfes wurde genutzt, um damit den momentanen Aufmerksamkeitsfokus des Nutzers abzuschätzen. Dieser Aufmerksamkeitsvektor wurde verwendet, um das Mikrofon mittels Softwaresteuerung für die Spracheingabe ein- beziehungsweise auszuschalten. Um das Mikrofon zu aktivieren, musste der Nutzer länger als eine Sekunde seinen Aufmerksamkeitsfokus auf den Kopf des Roboters richten. Damit der Benutzer feststellen konnte, ob das Mikrofon momentan eingeschaltet ist, wurden die im Kopf von ELIAS verbauten LEDs als visueller Rückmeldekanal in einen blinkenden Modus versetzt. Wenn das Mikrofon deaktiviert ist, leuchtet hingegen der LED-Ring dauerhaft. Dies ist auch in Abbildung 7.7 der Fall, da hier die Plattform auf einen Nutzer wartet.

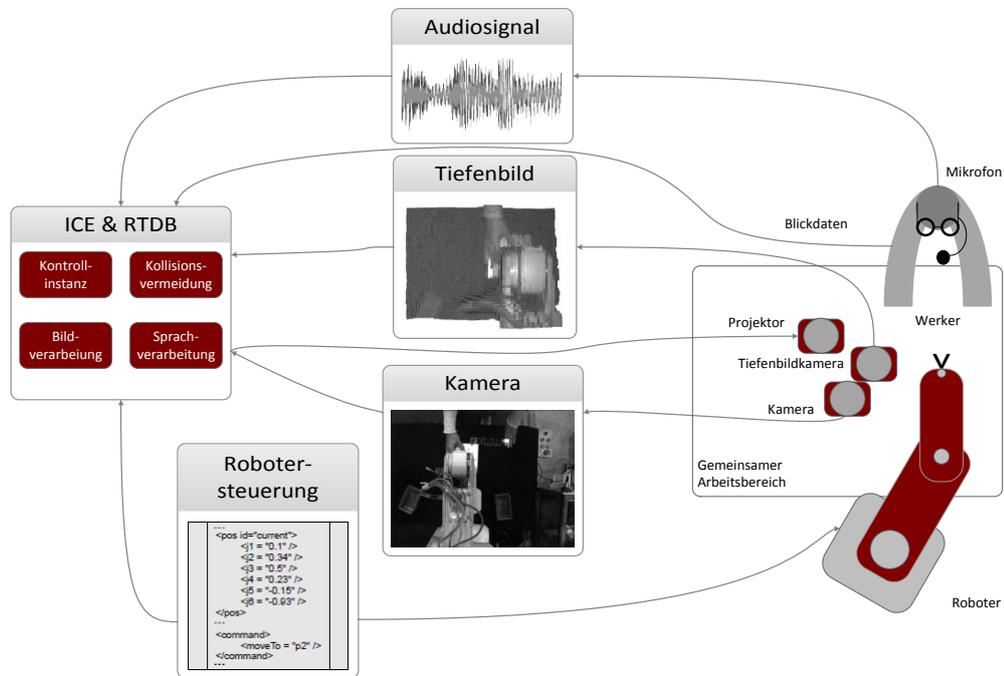
Das Spiel wurde in ähnlicher Art und Weise auf der Roboterplattform EDDIE implementiert. Auf dieser Plattform wurden damit unter anderem Studien zum Spiegeln von Gesichtsausdrücken [223] realisiert.

### **7.3 Industrielles Umfeld**

#### **7.3.1 Multimodale Mensch-Roboter-Interaktion in einer hybriden Fertigungszelle**

Der nachfolgend beschriebene Aufbau der hybriden Fertigungszelle ermöglichen eine multimodale Interaktion zwischen Mensch und Roboter sowie die gemeinsame Arbeit an der Montage eines Produkts. Die Komponenten für die hybride Fertigung mit der JAHIR-Forschungsplattform für diese Anwendung sind in Abbildung 7.8 dargestellt:

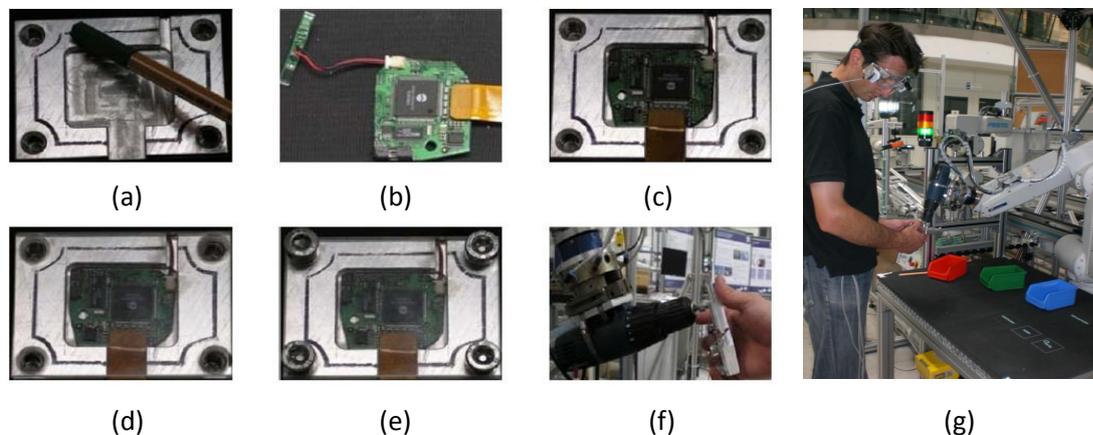
Die einzelnen Bauteile des Produkts sind eine Basisplatte, eine Leiterplatte, Verbindungskabel und eine Plastikabdeckung sowie vier Schrauben. Die Bauteile liegen zur besseren Erkennung und Handhabung durch den Roboter in farbigen Boxen vor. Diese Boxen befinden sich entweder bereits auf der Arbeitsfläche oder können durch eine Bandanbindung auf Paletten, die der Roboter vom Band nehmen kann, geliefert werden. Die Arbeitsschritte sind in Abbildung 7.9 (a) - (f) dargestellt. Für die Montage muss der Roboter zunächst die Kiste mit der Basisplatte überreichen. Der Werker kann dem System auf der Basisplatte mittels eines farbigen Zeigers (a) eine Klebelinie einlernen. Das System projiziert die erkannte Linie über den Projektor auf das Bauteil. Ist der Vorgang abgeschlossen, reicht der Roboter die Kiste mit dem Kabel



**Abbildung 7.8:** Überblick über die Komponenten für die Interaktion

und der Leiterplatte. Während der Roboter auf die Klebepistole wechselt und die Linie nachfährt, kann der Nutzer Kabel und Leiterplatte zusammenfügen, wie in (b). Ist der Roboter mit dem Klebevorgang fertig, fügt der Werker die Leiterplatte und das Kabel in die Ausbuchtung in der Basisplatte ein, wie in (c). In der Zwischenzeit holt der Roboter die Deckplatte und die Schrauben für Arbeitsschritt (d) und (e). Während der Werker die Platte aufsetzt und die Schrauben einsetzt, wechselt der Roboter auf den Akkuschauber. In Arbeitsschritt (f) zieht der Werker mit dem Akkuschauber am Roboter die Schrauben fest. Dem Nutzer werden dabei jeweils sowohl die passenden Anweisungen als auch die sensitiven Interaktionsflächen (siehe Abschnitt 4.3.6) auf der Arbeitsfläche dargeboten. Die sensitiven Interaktionsflächen können dabei entweder per Sprache, Hand oder Blick bedient werden.

Die Abbildung 7.9 (g) zeigt einen Werker beim letzten Arbeitsschritt, bei dem die Abdeckung mit dem Roboter auf die Basisplatte geschraubt wird. Hierbei kann er beispielsweise die Schraubrichtung des Akkuschaubers über die sensitiven Interaktionsflächen selektieren. Der Kraft-Momenten-Sensor wird verwendet, um die Übergabe der Boxen an den Werker zu detektieren. Außerdem kann damit die Geschwindigkeit des Akkuschaubers stufenlos geregelt werden. Die Farbkamera wird zur Detektion



**Abbildung 7.9:** Beispielprodukt: (a) - (f) Arbeitsschritte (g) Werker in hybrider Fertigungszelle bei Arbeitsschritt (f)

der Hände sowie der Boxen benutzt. Die Tiefenbildkamera kann die Segmentierung unterstützen [224] sowie für eine kollisionsvermeidende Bahnplanung [225] verwendet werden.

### 7.3.2 Intuitive Programmierung einer Verpackstation

Das Konzept für die intuitive Programmierung aus Abschnitt 6.4 wurde in einer Anwendung, in der ein Fernseher verpackt werden soll, im Rahmen einer Nutzerstudie bezüglich der Bedienbarkeit evaluiert. Nach einer kurzen Einführung in die Komponenten des CustomPacker-Systems und in das Programmierkonzept hatten die Probanden die Möglichkeit, den Ablauf selbst zu programmieren. Die Anzahl der Teilnehmer an der Nutzerstudie betrug 20 Personen. Von diesen Teilnehmern waren vier Personen weiblichen Geschlechts und 16 Probanden männlich. Das Alter der Versuchspersonen lag zwischen 22 und 40 Jahren.

85 % der Teilnehmer gaben an, viele oder sehr viele Kenntnisse im Umgang mit technischen Geräten zu haben. Diese hohe Zahl dürfte darauf zurückzuführen sein, dass inzwischen fast jeder im Rahmen von Smartphones, Tablets oder PCs mit Technik in Berührung kommt. 30 % der Probanden hatten gar keine und 25 % hatten kaum Erfahrung mit Industrierobotern.

Die konkrete Aufgabenstellung für die Probanden bestand darin, den letzten Teil eines Verpackprozesses zu programmieren, also die Schritte, nachdem der Roboter den Fernseher bereits gegriffen hat. Hierbei sollten Bereiche zur Überwachung durch eine Werkerpositionserkennung eingelernt, Positionen des Roboters abgespeichert sowie



**Abbildung 7.10:** Demonstratorzelle mit mobilem Eingabegerät zur Programmierung des Ablaufs

eine mögliche Geste zur Interaktion trainiert werden. Zu Beginn musste eine Warteposition für den Roboter definiert werden. Auf dieser Position sollte der Roboter warten, bis der Arbeiter den Karton für die Verpackung auf dem Förderband platziert hat. Für die Programmierung dieser Position konnten die Versuchsteilnehmer zwischen direktem Einlernen, natürlichsprachlichen Instruktionen und Editieren der Position direkt in einer visuellen Darstellung, wie exemplarisch in Abbildung 7.10 dargestellt, wählen.

Danach sollte der Roboter zunächst über und dann in den Karton fahren. Schließlich sollte der Roboter dann wieder aus dem Karton und zurück in die zuvor definierte Warteposition fahren. Der Zyklus wurde mit einer Geste des Nutzers abgeschlossen, dass alles in Ordnung ist. Hierfür hatte der Nutzer eine eigene Geste eintrainiert. Die so generierten Blöcke konnten dann in der visuellen Programmierumgebung dupliziert oder neu angeordnet werden. Wenn die Probanden ihr Programm als abgeschlossen betrachteten, konnten sie die Ausführung starten und den so programmierten Arbeitsablauf selbst beobachten und testen.

Im Anschluss daran wurden die Teilnehmer gebeten, einen erweiterten Fragebogen inklusive der beiden Evaluierungsmethoden (SUS und AttrakDiff) aus Abschnitt 3.3 auszufüllen. 90 % der Teilnehmer gaben an, dass es einfach bis sehr einfach für sie war, den Ablauf zu programmieren. Auch bei der Frage, wie gut sie insgesamt mit

## 7. ANWENDUNGEN

---

dem Programmierkonzept (Sprache, manuelles Führen und visuelle Programmierung) zurecht gekommen sind, gaben 22 % der Probanden an, gut und 72 % sogar sehr gut mit dem Konzept zurechtgekommen zu sein.

Hierfür spricht auch die Tatsache, dass alle Probanden die Aufgabe erfolgreich gelöst haben. Insbesondere auch Teilnehmer aus fachfremden Bereichen wie Betriebswirtschaftslehre, Marketing oder Übersetzung konnten die Aufgabe ohne große Schwierigkeiten bewältigen. Im Mittel über alle Teilnehmer benötigen sie für die Programmierung des kompletten Ablaufs 8,15 Minuten mit einer Standardabweichung von 3,66 Minuten. Auch diese recht kurze Zeit spricht für eine einfache Bedienbarkeit des entwickelten Programmierkonzepts. Die Fragen nach den einzelnen Modalitäten ergaben primär bei der Feinregelung der manuellen Führung des Roboters noch Verbesserungspotenzial. Dies ist wahrscheinlich auf die noch nicht ganz fertige Umsetzung des aktiven Kontaktflansches zur Messung der Kräfte in der vertikalen Achse zurückzuführen. Eventuell müssen auch die Parameter für die horizontale Positionierung noch nachjustiert werden.

Die korrekte Detektion der programmierten Bereiche war in den Testdurchläufen zu 95 % gegeben. Bei fünf Probanden wurde die tatsächliche Spracheingabe mitgeschrieben. Mittels der tatsächlichen Spracheingabe sowie der von der Spracherkennung gelieferten Texte kann die sogenannte Wortfehlerrate (engl.: Word Error Rate (WER)) berechnet werden. Dazu wird die Anzahl der notwendigen Schritte gezählt, um die vom Spracherkennung gelieferten Sätze in die tatsächlich gesprochenen umzuformen. Die Formel dazu lautet:

$$WER = \frac{S + D + I}{N} \quad (7.1)$$

Dabei bezeichnet  $S$  die Anzahl der Substitutionen,  $D$  die Anzahl der gelöschten Wörter,  $I$  die Anzahl neu einzufügender Wörter und  $N$  die Anzahl gesprochener Wörter. Für die hier präsentierte Beispielanwendung ergab sich eine mittlere Wortfehlerrate von 15,9 %.

Die trainierten Gesten wurden im getesteten System lediglich zu 40 % richtig erkannt. Hierbei ist allerdings zu erwähnen, dass die Probanden bei der Wahl der Geste im wahrsten Sinne des Wortes „vollkommen freie Hand“ hatten und das Training der Gesten nicht im sonst üblichen Laborumfeld stattfand. Des Weiteren wurden die Gesten nur einmal demonstriert.

Die Auswertung der SUS-Befragung (siehe Abschnitt 3.3.2) ergab ein überdurchschnittliches Ergebnis von  $\mu = 89,375$  Punkten im Mittelwert mit  $N = 20$  Teilnehmern

und einer Standardabweichung von  $\sigma = 6,38$  Punkten. Das Ergebnis spricht für eine intuitive und benutzerfreundliche Bedienbarkeit des Programmierkonzepts.

Des Weiteren wurden die Nutzer gefragt, ob sie die Spracheingabe lieber über eine Taste wie beim Telefon, über ein Schlüsselwort oder über Aufmerksamkeitsdetektion (beispielsweise auf eine bestimmte Stelle blicken) aktivieren wollen. Hierbei bevorzugten 65 % Prozent eine Taste für die Sprachbedienung, 25 % wollten gerne ein Schlüsselwort benutzen um die Spracherkennung zu starten und nur 10 % bevorzugten eine aufmerksamkeitsbasierte Aktivierung der Spracheingabe.



**Abbildung 7.11:** AttrakDiff-Ergebnisse der Nutzerstudie zum Programmierkonzept

Abbildung 7.11 zeigt die visuelle Auswertung der AttrakDiff-Ergebnisse der Nutzerstudie zum Programmierkonzept. Dabei repräsentiert *P* den Mittelwert aller abgegebenen Bewertungen und das helle orangefarbene Rechteck zeigt den Konfidenzbereich für die beiden Achsen hedonische und pragmatische Qualität an (siehe auch Abschnitt 3.3.1).

In diesem Fall ist das Konzept bei den Benutzern eindeutig als eher begehrt einzuordnen, da die Bewertung auch unter Berücksichtigung des Konfidenzbereichs noch innerhalb des Blocks liegt. Der recht schmale Konfidenzbereich spricht dafür, dass sich die Benutzer bei der Bewertung recht einig waren. Hinsichtlich der pragmatischen Qualität kann festgehalten werden, dass das entwickelte Konzept die Benutzer optimal unterstützt. Bei der hedonischen Qualität waren sich die Benutzer bei der Bewertung noch mehr einig und auch hier wird ein recht gutes Ergebnis erzielt. Die hedonische

## 7. ANWENDUNGEN

---

Qualität - Identität (also eine Art Markenbindung) kann noch verbessert werden, um die Benutzer stärker an das Produkt zu binden. Hinsichtlich der Stimulation bei der hedonischen Qualität liegt das Konzept über dem Durchschnitt und regt den Nutzer an und macht ihn neugierig. Insgesamt wirkt das Konzept auf die Benutzer aus hedonischer Sicht sehr attraktiv.

Des Weiteren kann somit festgehalten werden, dass sowohl SUS als auch Attrak-Diff zu dem übereinstimmenden Ergebnis einer überdurchschnittlichen Bedienbarkeit des entwickelten Systems kommen.

Außerdem wurden die Probanden auch nach ihrer subjektiv gefühlten Sicherheit in der Nähe des Industrieroboters gefragt. Hierbei gaben 95 % aller Teilnehmer an, sich sicher oder sogar sehr sicher gefühlt zu haben. Natürlich hatten die Probanden auch die Möglichkeit, Verbesserungsvorschläge einzureichen. Ein bei den Versuchen erkannter Punkt für Verbesserungen ist die Montageposition des Griffs für die manuelle Führung, da dieser insbesondere für kleinere Personen nicht immer ergonomisch zu erreichen war.

Insgesamt kamen 70 % der Probanden mit der manuellen Führung des Roboters gut oder sehr gut zurecht. Eine Möglichkeit zur Verbesserung ist hier die Feinanpassung der Regelparameter für die Gravitationskompensation des Roboters. Mit der Programmierung durch natürlichsprachliche Instruktionen kamen 40 % gut und 60 % der Probanden sogar sehr gut zurecht. Auch die visuelle Programmierumgebung fanden 95 % der Teilnehmer gut oder sehr gut.

Einige Probanden haben vom System erwartet, dass es automatisch erkennt, ob eine Geste, eine Position oder ein Bereich gespeichert werden soll und dann per Sprache nur den Bezeichner genannt. Mittels weiterer Techniken wäre es möglich, dies zum Teil automatisch zu erkennen. Wenn beispielsweise zeitnah der Roboter bewegt wurde, wird wahrscheinlich eine Roboterposition abgespeichert werden. Die Trennung zwischen Bereich und Geste ist ohne zusätzliche Kontextinformation schwieriger. Eine weitere Möglichkeit wäre hier, vorher eine Taste zu drücken, wenn man den Kontext nicht per Sprache vermitteln möchte.

Eine weitere Idee zur Verbesserung sind vordefinierte Gesten, aus denen die Benutzer auswählen können. Das Training von Gesten ist dann nur erforderlich, wenn die vorhandenen Gesten nicht mehr ausreichen oder der Benutzer eine andere Geste zur Interaktion bevorzugt. Eine grafische Visualisierung der aktuellen Position des Roboters in der GUI wurde von einem Probanden gefordert.

Schließlich wurde für die Simulation der Verwendung in einer Verpackstraße mit unterschiedlichen Produkten beziehungsweise Produktvarianten eine Fabrikchnittstelle implementiert. Durch diese Fabrikchnittstelle wird der Verpackstation mitgeteilt,



**Abbildung 7.12:** GUI zur Simulation der Fabrikschnittstelle

welches das nächste zu handhabende Produkt sein wird. In einer echten Verpackstraße würde das Signal mit der Information für das nächste Produkt beispielsweise aus dem Produktionssystem geliefert oder durch Scannen eines Barcodes auf der Palette, die das Packgut anliefert, erfasst werden. Für die Forschungsplattform kann mithilfe der GUI in Abbildung 7.12 das nächste Produkt der Verpackstraße ausgewählt und durch Anklicken der Schaltfläche *Signal senden* an die WEC geschickt werden. Außerdem wird in dieser GUI der Status der WEC angezeigt. Mit der Schaltfläche *Produktverzeichnis ändern* kann der Ablageort für die Informationen zu den Produkten verändert werden. Der vorab angelegte Arbeitsablauf für das Produkt ist in Form einer XML-Datei im Produktverzeichnis hinterlegt und wird durch die Schaltfläche *Signal senden* an die WEC übermittelt. Dadurch wird der Verpackprozess für dieses Produkt in der Verpackstation geladen und gestartet.

## 7. ANWENDUNGEN

---

## Zusammenfassung und Ausblick

Den Abschluss dieser Arbeit bilden eine kurze Zusammenfassung sowie ein Ausblick auf künftige Themen der Mensch-Roboter-Interaktion.

### 8.1 Zusammenfassung

In dieser Arbeit wurden entwickelte Konzepte für eine multimodale und intuitive Bedienung von Mensch-Roboter-Interaktionssystemen erläutert. Es wurde eine Architektur zur multimodalen Mensch-Roboter-Interaktion präsentiert sowie die entwickelten Komponenten zur Realisierung der Interaktionskanäle vorgestellt. Für die Steuerung der Abläufe wurde eine wissensbasierte Kontrollinstanz verwendet. Für Dialogsituationen zwischen Mensch und Roboter im sozialen Umfeld wurde eine Blicksteuerung konzipiert, die auf Studien zu menschlichem Blickverhalten basiert. Des Weiteren wurden Lernmethoden in der Mensch-Roboter-Kooperation hinsichtlich der Benutzerschnittstelle untersucht. In diesem Zusammenhang wurden Konzepte für instruktionsbasiertes Lernen in einer hybriden Fertigungszelle, das Lernen von Verpackungsschritten aus Punktwolken sowie eine intuitive Programmierung einer Verpackungstation untersucht.

Die prinzipielle Realisierbarkeit der Konzepte wurde durch eine anwendungsbezogene und prototypische Umsetzung auf Roboterplattformen und Forschungsplattformen demonstriert. Die gewählten Anwendungen wurden dabei sowohl im industriellen als auch im sozialen Umfeld realisiert. Des Weiteren wurden die Akzeptanz sowie die Bedienbarkeit der entwickelten Prototypen im Rahmen von Nutzerstudien untersucht. Hierbei zeigte sich im Allgemeinen eine positive Resonanz der Studienteilnehmer auf die entwickelten Konzepte.

Das Thema der Mensch-Roboter-Interaktion mit seiner faszinierenden Wirkung wurde in einem weiteren Projekt ausgewählt, um damit Schüler für das Thema zu motivieren. Dazu wurden exemplarisch Kurse für die Grundschule sowie für das Gymnasium entwickelt. Weitere Informationen zur Durchführung sowie zur Evaluierung des Projekts finden sich in [226].

### **8.2 Ausblick**

Der Bereich der Mensch-Roboter-Interaktion bietet noch viel Potenzial für zukünftige Forschungstätigkeiten. Die Sensoren und Methoden zur Erfassung des Menschen sowie seiner Sprache, seiner Aktionen, Intentionen und Emotionen werden immer ausgereifter und weniger invasiv [227]. Die dadurch anfallende Datenmenge muss analysiert und korrekt interpretiert werden. Wenn dies gelingt, werden sich zukünftige Roboter aufgrund dieser verbesserten Wahrnehmung noch besser an den Menschen und sein Umfeld anpassen können. Diese verbesserte Observation der menschlichen Aktivitäten wird sich ebenfalls positiv auf die Lernmethoden auswirken und zukünftige Systeme werden die durch den Benutzer demonstrierten Arbeitsschritte noch besser nachvollziehen können. Die Entwicklungen aus der Servicerobotik für den Einsatz im sozialen Umfeld werden wieder zurück in die Industrie fließen und dort neue Automatisierungsmöglichkeiten schaffen. Insgesamt wird die Servicerobotik immer mehr mit der Industrierobotik verschmelzen [228]. Neue Eingabegeräte, wie intelligente Uhren oder Brillen, bieten neue Interaktionsmöglichkeiten für die Zusammenarbeit mit Robotern. Ein weiteres interessantes Gebiet sind die Entwicklungen von neuronalen Schnittstellen für die Interaktion, die heutzutage beispielsweise schon die Steuerung von künstlichen Extremitäten durch Gedanken ermöglichen [229]. Zukünftig werden auch die manipulierten Objekte in der Industrie Informationen mit sich tragen. Zu diesem Zweck wurde ein Konsortium [230] gegründet, um herstellerübergreifende Standards für den Datenaustausch im Internet der Dinge zu etablieren. Dadurch könnten die Objekte dem Roboter beispielsweise gleich die Manipulationsdaten (Greifpunkte im CAD-Modell) sowie die Montagedaten bereitstellen. In [197] wird der Ansatz verfolgt, dass Robotersysteme gesammelte Erfahrungen und erworbenes Wissen über das Internet austauschen und übertragen können.

# Glossar

## **Aktoren**

Aktoren dienen zur Ausführung von Aktionen in der Umgebung. Die Signale dazu werden von der Steuerung empfangen und in Aktionen (beispielsweise mechanische Bewegung) in der Umgebung umgesetzt.

## **Arbeitsablauf**

Definierte Abfolge von Arbeitsschritten, um eine Aufgabe zu erledigen.

## **Arbeitsschritt**

Schritt in einem Arbeitsablauf zur Erfüllung der Aufgabe.

## **Avatare**

Künstliche Erscheinungsbilder meist zur Repräsentation echter Personen in der virtuellen Welt.

## **Bayessches Netz**

Azyklischer gerichteter Graph, dessen Knoten Variablen und dessen Kanten bedingte Abhängigkeiten zwischen diesen Variablen darstellen.

## **Forschungsplattform**

Die Forschungsplattform umfasst alle Komponenten des gesamten Demonstratoraufbaus. Dazu gehören neben der Roboterplattform auch externe Aktoren, Sensoren sowie weitere Aufbauten.

## **Intuitive Interaktion**

Interaktion, die auf Anhieb gelingt, ohne die grundlegenden Zusammenhänge kennen zu müssen und daher keine Lernphase benötigt.

**Kontrollinstanz**

Die Kontrollinstanz überwacht und regelt die Ausführung des Ablaufs. In der Industrie ist die Kontrollinstanz für die Arbeitsablaufsteuerung zuständig und im sozialen Umfeld regelt die Kontrollinstanz den Dialog beziehungsweise die Interaktion.

**Lernmodus**

Im Lernmodus kann der Anwender das System programmieren und dadurch unter anderem den Arbeitsablauf erstellen oder anpassen.

**Mensch-Roboter-Interaktion**

Kommunikation beziehungsweise Informationsaustausch zwischen Mensch und Roboter.

**Mensch-Roboter-Kooperation**

Eine Art der Zusammenarbeit zwischen Mensch und Roboter mit dem Hintergrund, durch die Interaktion zwischen Mensch und Roboter ein gemeinsames Ziel zu erreichen.

**Modalität**

Ein- oder Ausgabekanal zur Wahrnehmung oder Interaktion mit der Umgebung.

**Multimodalität**

Einsatz verschiedener Modalitäten in Systemen zur Interaktion.

**Packgut**

die eigentlich zu verpackende Ware.

**Packmittel**

Art der Verpackung, wie zum Beispiel Dose, Kiste, Schachtel

**Produktivmodus**

Im Produktivmodus folgt das System dem aktuellen Arbeitsablauf, um das Produkt fertigzustellen.

**Proxy-Server**

Stellvertretender Server, der im Netzwerk für zwei weitere Kommunikationsteilnehmer den jeweils anderen repräsentiert und somit auf die Inhalte beider Kommunikationsteilnehmer zugreifen kann.

**Publisher**

Programmteil, der Informationen unter einem bestimmten Thema beim Topic-Manager veröffentlicht. Dieses Thema kann von anderen Programmen abonniert werden, wodurch diese immer die aktuellen Informationen zu diesem Thema erhalten.

**Punktwolke**

Menge von Punkten mit expliziten x-, y- und z-Daten sowie möglicherweise zusätzlichen Informationen zu Farbe, Normalen etc.

**Roboterplattform**

Roboter zusammen mit den Sensoren, Aktoren sowie weiteren Erweiterungen, die direkt am Roboter befestigt sind.

**Sakkaden**

Gleichsinnige und schnelle Bewegung beider Augen. Diese Augenbewegung kann dabei auf Kommando, willkürlich, spontan sowie zur Erfassung eines Blickziels erfolgen.

**Salienz**

Auffälligkeit eines Reizes, der meist aus seinem Umfeld heraussticht.

**Schlüsselszene**

Eine Schlüsselszene ist eine für die weitere Verarbeitung besonders relevante Aufnahme einer Situation.

**Sensoren**

Sensoren dienen zur Wahrnehmung der Umgebung. Die Sensoren wandeln dazu die erfasste Information in elektrische Signale um und leiten diese Signale zur Weiterverarbeitung an eine Steuerung weiter.

**Subscriber**

Programmteil, der Informationen, die zu einem bestimmten Thema veröffentlicht wurden, beim TopicManager abonniert. Der Subscriber erhält dadurch die aktuellen Informationen zu diesem Thema.

**Teilautomat**

Zustandsautomat, der selbst einen Teil eines größeren Zustandsautomaten bildet.

**Tiefenbild**

Zweidimensionales Bild, dessen Pixelwerte für die Tiefe stehen. Üblicherweise wird der Abstand nah zu fern mit der Intensität codiert. Die x-, y- und z-Daten sind implizit hinterlegt und lassen sich aus der Pixelposition im Bild, dem Intensitätswert des Pixels und der Kamerageometrie berechnen.

**TopicManager**

Programm, das der Vermittler zwischen Publisher und Subscriber ist. Sowohl der Publisher als auch der Subscriber registrieren sich bei diesem Programm. Der TopicManager übernimmt die Informationsverteilung.

**Verpackprozess**

Die notwendigen Verpackschritte im Arbeitsablauf zur Verpackung eines Packguts.

**Verpackschritt**

Erforderlicher Schritt in einem Verpackprozess zur Verpackung des Packguts.

**Voxel**

Zusammensetzung aus Volumen und Pixel. Repräsentiert ein volumenhaftes Pixel im dreidimensionalen Raum.

**WebSocket**

Auf TCP basierendes Netzwerkprotokoll zur bidirektionalen Verbindung einer Anwendung im Web mit einem WebSocket-Server.

**Zykluszeit**

Die Zykluszeit bezeichnet in der industriellen Produktion die benötigte Dauer für den Durchlauf eines kompletten Arbeitsablaufs der Station.

# Abkürzungsverzeichnis

<i>iwb</i>	Institut für Werkzeugmaschinen und Betriebswissenschaften
API	Application Programming Interface
ARTK	Augmented Reality Tool Kit
AUI	Attentive User Interface
BMBF	Bundesministerium für Bildung und Forschung
BNF	Backus-Naur-Form
CAD	Computer-Aided Design
CCRL	CoTeSys Central Robotics Laboratory
CLI	Command Line Interface
CoTeSys	Cognition for Technical Systems
CQD	Chi-Quadrat-Distanz
CustomPacker	Highly Customizable and Flexible Packaging Station for mid- to upper sized Electronic Consumer Goods using Industrial Robots
DFG	Deutsche Forschungsgemeinschaft
EDDIE	Emotion-Display with Dynamic Intuitive Expressions
EDF	Earliest Deadline First
EER	Equal Error Rate
ELIAS	Enhanced Living Assistant

## Abkürzungsverzeichnis

---

FPFH	Fast Point Feature Histogram
GUI	Graphical User Interface
HMM	Hidden Markov Model
HTML	Hyper Text Markup Language
IBL	Instruktionsbasiertes Lernen
ICE	Internet Communication Engine
ICP	Iterative Closest Point
IFF	Institut für Fabrikbetrieb und -automatisierung
IFR	International Federation of Robotics
IPA	Institut Produktionstechnik und Automatisierung
ISI	Institut für System- und Innovationsforschung
ITD	Interaural Time Difference
JAHIR	Joint Action for Humans and Industrial Robots
JESS	Java Expert System Shell
LISP	LISt Processing
LoA	Level of Automation
LSR	Lehrstuhl für Steuerungs- und Regelungstechnik
MPEG	Moving Pictures Expert Group
MRK	Mensch-Roboter-Kooperation
MuDiS	Multimodal Dialog System
NTP	Network Time Protocol
NUI	Natural User Interfaces
PCL	Point Cloud Library
PFH	Point Feature Histogram
RANSAC	Random Sample Consensus
ROC	Receiver Operating Characteristic
ROS	Robot Operating System
RPC	Remote Procedure Call
RTDB	Real-Time Database
SAC-IA	Sample Consensus Initial Alignment
SDK	Software Development Kit

Siri	Speech Interpretation and Recognition Interface
SOAP	Simple Object Access Protocol
SPFH	Simplified Point Feature Histogram
SUN	Saliency Using Natural statistics
SUS	System Usability Scale
TCP	Transmission Control Protocol
TCP	Tool Center Point
TUI	Tangible User Interface
UDP	User Datagram Protocol
VFH	Viewpoint Feature Histogram
VOG	Videookulographie
WEC	Workflow Execution Control
WER	Word Error Rate
WIMP	Windows Icons, Menus, Pointer
YARP	Yet Another Robot Plattform
ZV	Zufallsvariable

## Abkürzungsverzeichnis

---

# Abbildungsverzeichnis

1.1	Produktvielfalt am Beispiel eines Loewe Fernsehers Individual [3] . . .	2
1.2	Tablet-Steuerung von Robotern . . . . .	3
2.1	Roboterentwicklung . . . . .	8
2.2	Gemeinsamer Arbeitsbereich von Werker und Roboter in der hybriden Montage aus [30] . . . . .	11
2.3	Kombinationen aus zeitlicher und örtlicher Trennung [30] . . . . .	11
2.4	Forschungsansätze für den Einsatz von Robotern in der Nähe des Menschen . . . . .	12
2.5	Assistenzroboter Baxter von Rethink Robotics [41] . . . . .	14
2.6	Sensorgesteuerte Fertigung mit dem Fertigungsassistent [55] . . . . .	18
3.1	Erzeugung des Tiefenbildes mithilfe eines projizierten Musters . . . . .	20
3.2	Winkelmerkmalsberechnung für das PFH, entnommen aus [60]. . . . .	21
3.3	Blickpunktabhängiges Winkelmerkmal für das VFH [61] . . . . .	22
3.4	FPFH Komponente für den Schwerpunkt $\mathbf{c}$ [61] . . . . .	23
3.5	HMM mit $N = 3$ versteckten Zuständen . . . . .	25
3.6	Teachpanel KeTopT50 der Firma KEBA [75] . . . . .	28
3.7	Exemplarische Einzelauswertung für ein Produkt mittels AttrakDiff [83] . . . . .	31
4.1	Modalitäten zwischen Mensch und Maschine . . . . .	39
4.2	Roboterkopf EDDIE [119] zeigt sechs Basisemotionen [122]. . . . .	43
4.3	EyeSeeCam [125] . . . . .	45
4.4	Drei auf die Arbeitsoberfläche projizierte sensitive Interaktionsflächen . . . . .	46
4.5	Bilder aus der Kopfgestensequenz Nicken . . . . .	47

## ABBILDUNGSVERZEICHNIS

---

4.6	Prinzip der Kommunikation der Sensoren sowie Aktoren mit der wissensbasierten Kontrollinstanz über ICE und RTDB . . . . .	50
4.7	Ansteuerung der Aktoren aufgrund von Schlussfolgerungen in der Wissensbasis basierend auf den gelieferten Fakten der Sensoren . . . . .	51
4.8	Datenverwaltung und Speicherung in der RTDB, vergleiche [88] . . . . .	52
5.1	Exemplarische Top-Down und Bottom-Up Einflüsse auf das Blickverhalten in Dialogsituationen . . . . .	56
5.2	Roboter mit Augensteuerung . . . . .	58
5.3	Konzept für die Steuerung des Blickverhaltens . . . . .	60
5.4	Blickmodell für die Übergänge zwischen den Beobachtungen . . . . .	61
5.5	Parametrisiertes CANDIDE-3-Gesichtsmodell [167], berechnet mittels Microsoft Kinect für Windows Gesichtsverfolgungs-SDK . . . . .	63
5.6	Interaurale Zeitdifferenz . . . . .	64
5.7	Farbbild und zugehörige Salienzkarte skaliert auf $160 \times 120$ Pixel . . . . .	66
5.8	Blickmodell mit den Transitionswahrscheinlichkeiten zwischen den relevanten Beobachtungen für den Dialogmodus Sprecher . . . . .	67
5.9	Sensor, Roboterkopf und -augen sowie Gesichtsmodell in der Seitenansicht [174] . . . . .	68
5.10	Ergebnisse zu den Fragen bezüglich der Gesichtsmerkmale . . . . .	70
5.11	Ergebnisse zur visuellen Salienz . . . . .	71
5.12	Ergebnisse zur auditiven Salienz . . . . .	72
5.13	Ergebnisse der allgemeinen Fragen . . . . .	73
6.1	Simulationssoftware KUKA Sim [185] zur Offline-Programmierung der Roboter . . . . .	78
6.2	Sicheres und direktes Teach-In der Firma MRK-Systeme GmbH [188] . . . . .	79
6.3	Demonstrationsverfahren für Roboter . . . . .	82
6.4	Wissensbasierte Kontrollinstanz mit Sensoren und Aktoren sowie generiertem Ablauf . . . . .	85
6.5	Aktionen zur Verpackschritterkennung aus der Punktwolke. . . . .	87
6.6	Bildverarbeitungsschritte zur Markerdetektion mit ARTK [200] . . . . .	89
6.7	Transformierte Punktwolke in das Marker-Koordinatensystem. . . . .	90
6.8	Oktalbaum: Die linke Seite zeigt die Zerlegung des Ausgangsvolumens $A$ in je acht Würfel. Die Repräsentation als Oktalbaum ist auf der rechten Seite dargestellt. Vergleiche [201] . . . . .	91
6.9	Voxelbasierte Darstellung einer Punktwolke, die als Oktalbaum repräsentiert ist [60]. . . . .	92

6.10	Gefilterte Punktwolke . . . . .	93
6.11	Segmentierung der Tischebene aus $\mathcal{P}_f$ . . . . .	94
6.12	Begrenzungslinie zur Schlüsselszenenerkennung . . . . .	100
6.13	Sensorposen auf der Roboterplattform . . . . .	101
6.14	iPad-Verpackprozess . . . . .	102
6.15	Spiegelungen bei Objekten (iPad, Mac-Mini, Spiegel): Links RGB-Bild, rechts Tiefenbild . . . . .	103
6.16	Konzept für die intuitive Programmierung der Verpackstation . . . . .	104
6.17	GUI-Konzept im CustomPacker. Visuelle Programmieroberfläche, basierend auf Blockly [207], mit den Blöcken für den Arbeitsablauf. . . . .	105
6.18	CustomPacker-WEC-Applikation . . . . .	106
6.19	Erzeugter Block aus natürlichsprachlicher Instruktion für die visuelle Repräsentation des Arbeitsablaufs . . . . .	107
6.20	Personenhypothesen . . . . .	108
6.21	Skelettstruktur mit Winkeln . . . . .	108
7.1	Roboterplattform ELIAS: . . . . .	112
7.2	Mobile Roboterplattform des LSR im CCRL mit Kopf EDDIE [119] . . . . .	113
7.3	Werker bei der Schraubmontage in der hybriden Fertigungszelle JAHIR . . . . .	114
7.4	Werker im Produktivmodus in der CustomPacker Prototyp-Zelle . . . . .	115
7.5	Links: Aufbau der Laborumgebung im CCRL [219]; rechts: Dialogfluss für die Getränkebestellung . . . . .	117
7.6	EDDIE und Gast im Dialog . . . . .	118
7.7	Komponenten und Informationsfluss auf der Plattform ELIAS . . . . .	119
7.8	Überblick über die Komponenten für die Interaktion . . . . .	121
7.9	Beispielprodukt: (a) - (f) Arbeitsschritte (g) Werker in hybrider Fertigungszelle bei Arbeitsschritt (f) . . . . .	122
7.10	Demonstratorzelle mit mobilem Eingabegerät zur Programmierung des Ablaufs . . . . .	123
7.11	AttrakDiff-Ergebnisse der Nutzerstudie zum Programmierkonzept . . . . .	125
7.12	GUI zur Simulation der Fabrikschnittstelle . . . . .	127

## ABBILDUNGSVERZEICHNIS

---

# Tabellenverzeichnis

2.1	Automatisierungslevel (engl: Level of Automation, LoA) [29] für Werkzeuge und Kontrollmodus . . . . .	10
4.1	Mittelwerte der sechsfachen Kreuzvalidierung für die Kopfgestenerkennung mit kontinuierlichen HMMs [134]. . . . .	48
6.1	Erkennungsergebnisse im Zusammenhang mit der Anzahl der Punkte für die Objekte . . . . .	100
6.2	Gewicht und Bereiche der Winkel in der Skelettstruktur . . . . .	109
6.3	Konfusionsmatrix für Personenverifikation mit Schwellwert 0,4 . . . .	109



# Literaturverzeichnis

- [1] INTERNATIONAL FEDERATION OF ROBOTICS. **Robots Create Jobs**, 2011. 1
- [2] UGEBREVET A4. **Robotter øger dansk konkurrenceevne (Roboter erhöhen die dänische Wettbewerbsfähigkeit)**, 2012. 1
- [3] LOEWE AG. **Individualisierung**. 2, 77, 139
- [4] M. HÄGELE, N. BLÜMLEIN, UND O. KLEINE. **Wirtschaftlichkeitsanalysen neuartiger Servicerobotik-Anwendungen und ihre Bedeutung für die Robotik-Entwicklung**, 2011. 2
- [5] REIS GMBH & Co. KG MASCHINENFABRIK . **ReisPAD**, 2012. 3, 4
- [6] UNIVERSAL ROBOTS. **Produktimpressionen**. 3, 4
- [7] THE HISTORY OF COMPUTING PROJECT. **Timeline of Robotics**. 7
- [8] K. ČAPEK. **R.U.R.** 7
- [9] I. ASIMOV. **Runaround**. 7
- [10] INTERNATIONAL FEDERATION OF ROBOTICS. **History of Industrial Robots – From the first installation until today**. 7, 8
- [11] KUKA ROBTER GMBH. **Historie – Erste Schweisstransferstrasse Europas mit Robotern**. 7
- [12] KUKA ROBTER GMBH. **KR 1000 TITAN**. 7, 8
- [13] VDI. **VDI-Richtlinie 2860: Montage- und Handhabungstechnik; Handhabungsfunktionen, Handhabungseinrichtungen; Begriffe, Definitionen, Symbole**. VDI-Gesellschaft Produktionstechnik, 1990. 8
- [14] YASKAWA AMERICA. **Motoman SDA10 robot assembly**. 8
- [15] FRAUNHOFER IPA. **Care-O-Bot**. 8, 37
- [16] INTERNATIONAL FEDERATION OF ROBOTICS. **More than 160000 industrial robots sold in 2012**, 2013. 8

## LITERATURVERZEICHNIS

---

- [17] AUDI AG. **Karosseriebau beim Audi A3.** 8
- [18] ROTH & RAU - ORTNER GMBH. **SCOUT – Mobile Robot for Efficient & Flexible Use in Cleanrooms.** 8
- [19] PERCEPTRON GMBH. **Chassis-Messungen.** 8
- [20] ALDEBARAN ROBOTICS. **NAO.** 8, 37
- [21] LEGO. **Mindstorms.** 8, 37
- [22] BOSTON DYNAMICS. **BigDog - The Most Advanced Rough-Terrain Robot on Earth.** 8
- [23] IROBOT. **Roomba.** 8, 37
- [24] Y.-H. CHOI, J.-Y. LEE, J.-D. LEE, UND K.-E. LEE. **SMART WINDORO V1.0: Smart window cleaning robot.** In *Ubiquitous Robots and Ambient Intelligence (URAI), 2012 9th International Conference on*, Seiten 116–119, Nov 2012. 8
- [25] M. FISCHETTI. **Der vollautomatische Gärtner : Rasenmäher ; Robbie mäht den Rasen.** *Spektrum der Wissenschaft*, Seiten 94–95, 2008. 8
- [26] T. MUKAI, S. HIRANO, H. NAKASHIMA, Y. KATO, Y. SAKAIDA, S. GUO, UND S. HOSOE. **Development of a nursing-care assistant robot RIBA that can lift a human in its arms.** In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, Seiten 5996–6001, Oct 2010. 9
- [27] C. BORST, T. WIMBÖCK, F. SCHMIDT, M. FUCHS, B. BRUNNER, F. ZACHARIAS, P. R. GIOR-DANO, R. KONIETSCHKE, W. SEPP, S. FUCHS, C. RINK, A. ALBU-SCHÄFFER, UND G. HIR-ZINGER. **Rollin’ Justin - Mobile platform with variable base.** In *International Conference on Robotics and Automation (ICRA)*, Seiten 1597–1598. IEEE, 2009. 9
- [28] Y. LI, C. LI, UND P. CHEN. **Research and design of control system for a tracked SAR robot under coal mine.** In *IEEE International Conference on Automation and Logistics (ICAL), 2009.*, Seiten 1957–1961, Aug 2009. 9
- [29] J. FROHM. **Levels of Automation in Production Systems.** Dissertation, CHALMERS UNIVERSITY OF TECHNOLOGY, Göteborg, Schweden, 2008. 9, 10, 143
- [30] S. THIEMERMANN. **Direkte Mensch-Roboter-Kooperation in der Kleinteilemontage mit einem SCARA-Roboter.** Dissertation, Universität Stuttgart, 2005. 10, 11, 139
- [31] J. DE WINTER UND D. DODOU. **Why the Fitts list has persisted throughout the history of function allocation.** *Cognition, Technology & Work*, Seiten 1–11, August 2011. 10
- [32] FESTO AG & CO. KG. **Der Bionische Handling-Assistent – nachgiebig und frei beweglich.** 12
- [33] R. DAHIYA, P. MITTENDORFER, M. VALLE, G. CHENG, UND V. LUMELSKY. **Directions Toward Effective Utilization of Tactile Skin: A Review.** *Sensors Journal, IEEE*, **13**(11):4121–4138, Nov 2013. 12

- [34] M. FRITZSCHE, N. ELKMANN, UND E. SCHULENBURG. **Tactile sensing: a key technology for safe physical human robot interaction.** In *Proceedings of the 6th international conference on Human-robot interaction, HRI '11*, Seiten 139–140, New York, NY, USA, 2011. ACM. 12
- [35] P. MITTENDORFER. **HEX-o-SKIN – From a multi-modal intelligent cell to a self-organising robotic skin.** 12
- [36] M. GIULIANI, C. LENZ, T. MÜLLER, M. RICKERT, UND A. KNOLL. **Design Principles for Safety in Human-Robot Interaction.** *International Journal of Social Robotics*, Seiten 253–274, 2010. 12
- [37] EUROPEAN COMMITTEE FOR STANDARDIZATION (CEN). **ISO 10218-1: Robots and robotic devices - Safety requirements for industrial robots - Part 1: Robots.**, 2011. 12
- [38] EUROPEAN COMMITTEE FOR STANDARDIZATION (CEN). **ISO 10218-2: Robots and robotic devices - Safety requirements for industrial robots - Part 2: Robot systems and integration.**, 2011. 12
- [39] S. BECK. **Jenseits von Mensch und Maschine: Ethische und rechtliche Fragen zum Umgang mit Robotern, Künstlicher Intelligenz und Cyborgs.** Robotik und Recht. Nomos Verlagsgesellschaft mbH und Co. KG, 2012. 12
- [40] T. FONG, I. R. NOURBAKHS, UND K. DAUTENHAHN. **A survey of socially interactive robots.** *Robotics and Autonomous Systems*, **42**(3-4):143–166, 2003. 13
- [41] RETHINK ROBOTICS. **Baxter – A unique robot with unique features**, 2012. 14, 139
- [42] T. BRODTMANN. **Google kann Service-Robotik in Fahrt bringen**, Januar 2014. 15
- [43] C. LENZ, M. RICKERT, G. PANIN, UND A. KNOLL. **Constraint task-based control in industrial settings.** In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, Seiten 3058 –3063, oct. 2009. 15
- [44] M. HUBER, M. RICKERT, A. KNOLL, T. BRANDT, UND S. GLASAUER. **Human-robot interaction in handing-over tasks.** In *Robot and Human Interactive Communication, 2008. RO-MAN 2008. The 17th IEEE International Symposium on*, Seiten 107 –112, aug. 2008. 15
- [45] C. WENDT, M. POPP, UND B. FAERBER. **Emotion induction during human-robot interaction.** In *Proceedings of the 4th ACM/IEEE international conference on Human robot interaction, HRI '09*, Seiten 213–214, New York, NY, USA, 2009. ACM. 15
- [46] EC - SIXTH FRAMEWORK PROGRAMME (FP6). **COGNIRON - The Cognitive Robot Companion.** 16
- [47] BAYERISCHE FORSCHUNGSSTIFTUNG. **EsIMiP - Efficient and safe interaction of humans and intelligent production systems.** 16
- [48] UNIVERSITÄT BAYREUTH. **INTROP – Intuitive Programmierung von Roboter-Manipulatoren.** 17

## LITERATURVERZEICHNIS

---

- [49] BUNDESMINISTERIUM FÜR BILDUNG UND FORSCHUNG (BMBF). **LiSA - Assistenzroboter in Laboren von Life-Science-Unternehmen.** 17
- [50] EC - SEVENTH FRAMEWORK PROGRAMME (FP7). **LoCoBot - Low Cost Robot Co-Workers.** 17
- [51] EC - SIXTH FRAMEWORK PROGRAMME (FP6). **PiSA - Flexible Assembly Systems through Workplace-Sharing and Time-Sharing Human-Machine Cooperation.** 17
- [52] EC - SEVENTH FRAMEWORK PROGRAMME (FP7). **PRACE - The Productive Robot Apprentice.** 18
- [53] EC - SEVENTH FRAMEWORK PROGRAMME (FP7). **SMErobotics.** 18
- [54] KUKA ROBOTER GMBH. **LBR iiwa.** 18
- [55] M. NAUMANN UND R. AWAD. **Assembly Assistant (Fraunhofer IPA).** 18, 139
- [56] M. WEGNER, M. KÜCKELHAUS, K. RICHER, O. POENICKE, S. STECK, UND F. ORTMANN. **Low-Cost Sensor Technology – A DHL perspective on implications and use cases for the logistics industry,** Dezember 2013. 20
- [57] SIEMENS AG. **Kinect-Sensor entschärft gefährliche Arbeitsabläufe,** 2013. 20
- [58] B. FREEDMAN, A. SHPUNT, M. MACHLINE, UND Y. ARIELI. **Depth mapping using projected patterns,** 2010. Patent-Nr.: US 2010/0118123 A1, PrimeSense LTD. 20
- [59] K. KHOSHESHAM. **Accuracy analysis of kinect depth data.** In *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences (ISPRS) workshop laser scanning*, **38**, Seiten 133–138, 2011. 20
- [60] R. B. RUSU. **Semantic 3D Object Maps for Everyday Manipulation in Human Living Environments.** Dissertation, Fakultät für Informatik, Technische Universität München, Oktober 2009. 21, 22, 90, 92, 139, 140
- [61] R. B. RUSU, G. BRADSKI, R. THIBAU, UND J. HSU. **Fast 3D Recognition and Pose Using the Viewpoint Feature Histogram.** In *Proceedings of the 23rd IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Taipei, Taiwan, Oktober 2010. 22, 23, 94, 139
- [62] L. RABINER. **A tutorial on hidden Markov models and selected applications in speech recognition.** *Proceedings of the IEEE*, **77(2):257–286**, Feb 1989. 24
- [63] J. SCHENK. **Online-Erkennung handgeschriebener Whiteboard-Notizen.** Dissertation, Technische Universität München, München, 2009. 24
- [64] F. WALLHOFF. **Entwicklung und Evaluierung neuartiger Verfahren zur automatischen Gesichtsdetektion, Identifikation und Emotionserkennung.** Dissertation, Technische Universität München, 2006. 24, 117

- [65] A. WILSON UND A. BOBICK. **Parametric hidden Markov models for gesture recognition.** *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **21**(9):884–900, Sep 1999. 24
- [66] T. INAMURA, H. TANIE, UND Y. NAKAMURA. **Keyframe compression and decompression for time series data based on the continuous hidden Markov model.** In *Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, **2**, Seiten 1487 – 1492 vol.2, oct. 2003. 24, 80
- [67] B. SICILIANO UND O. KHATIB, Editoren. **Springer Handbook of Robotics.** Springer, Berlin, Heidelberg, 2008. 24, 80, 82
- [68] G. D. FORNEY JR. **The viterbi algorithm.** *Proceedings of the IEEE*, **61**(3):268–278, 1973. 25
- [69] M. A. FISCHLER UND R. C. BOLLES. **Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography.** *Communications of the ACM*, **24**:381–395, June 1981. 26, 93
- [70] P. H. S. TORR UND A. ZISSERMAN. **MLESAC: A New Robust Estimator with Application to Estimating Image Geometry.** *Computer Vision and Image Understanding*, **78**:2000, 2000. 26
- [71] J. SCHENK UND G. RIGOLL. **Mensch-Maschine-Kommunikation**, 2010. 27
- [72] N. J. NILSSON. **Shakey The Robot.** Technischer Bericht 323, AI Center, SRI International, 333 Ravenswood Ave., Menlo Park, CA 94025, April 1984. 27
- [73] J. MCCARTHY. **Recursive Functions of Symbolic Expressions and Their Computation by Machine, Part I.** *Communications of the ACM*, **3**(4):184–195, April 1960. 27
- [74] R. R. KUMAR. **Human Computer Interaction.** Laxmi Publications, 2005. 27
- [75] KEBA. **KeTop T50 VGA zum Trend100-Produkt gekürt.** 27, 28, 139
- [76] C. GUO UND E. SHARLIN. **Exploring the use of tangible user interfaces for human-robot interaction: a comparative study.** In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Seiten 121–130. ACM, 2008. 28
- [77] R. VERTEGAAL. **Attentive user interfaces.** *Communications of the ACM*, **46**(3):30–33, 2003. 28
- [78] MICROSOFT. **Kinect for Windows Human Interface Guidelines.** 29
- [79] C. R. BURGHART UND A. STEINFELD. **Proceedings of Metrics for Human-Robot Interaction, a Workshop at ACM/IEEE HRI 2008.** Technischer Bericht 323, University of Hertfordshire, March 2008. 29
- [80] M. RUDINAC, J. R. DE SOUZA, R. XIONG, P. SMITHMAITRIE, M. MATAMOROS, UND V. SEIB. **RoboCup@Home.** 29
- [81] W. NOWAK, A. HITZMANN, F. HEGGER, J.-D. FOSSEL, UND R. PEGORARO. **RoboCup@Work.** 29

## LITERATURVERZEICHNIS

---

- [82] S. V. STEHMAN. **Selecting and interpreting measures of thematic classification accuracy.** *Remote sensing of Environment*, **62**(1):77–89, 1997. 30
- [83] USER INTERFACE DESIGN GMBH. **AttrakDiff**. 31, 139
- [84] M. HASSENZAHL, M. BURMESTER, UND F. KOLLER. **Der User Experience (UX) auf der Spur: Zum Einsatz von www.attrakdiff.de.** Usability Professionals 2008, Stuttgart, 2008. 31
- [85] J. R. LEWIS UND J. SAURO. **The Factor Structure of the System Usability Scale.** In *Proceedings of the 1st International Conference on Human Centered Design: Held as Part of HCI International 2009*, HCD 09, Seiten 94–103, Berlin, Heidelberg, 2009. Springer-Verlag. 32
- [86] A. BANGOR, P. T. KORTUM, UND J. T. MILLER. **An Empirical Evaluation of the System Usability Scale.** *International Journal of Human-Computer Interaction*, **24**:574 – 594, 2008. 32
- [87] ZERO.C. **Internet Communication Engine**. 32, 38
- [88] M. GOEBL. **Eine realzeitfähige Architektur zur Integration kognitiver Funktionen.** Dissertation, Technische Universität München, München, 2009. 33, 52, 140
- [89] M. GOEBL. **KogMo-RTDB - Real-time Database for Cognitive Automobiles**, 2011. 33
- [90] R. B. RUSU. **Point Cloud Library**. 34, 88
- [91] R. B. RUSU UND S. COUSINS. **3D is here: Point Cloud Library (PCL).** In *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, May 9-13 2011. 34
- [92] R. B. MILLER. **Response Time in Man-computer Conversational Transactions.** In *Proceedings of the December 9-11, 1968, Fall Joint Computer Conference, Part I*, American Federation of Information Processing Societies (AFIPS) '68 (Fall, part I), Seiten 267–277, New York, NY, USA, 1968. ACM. 35
- [93] E. PRASSLER, R. BISCHOFF, UND H. BRUYNINCKX. **It's a Long Way to Useful Standards [Industrial Activities].** *Robotics Automation Magazine, IEEE*, **17**(3):18–19, 2010. 36
- [94] G. METTA, P. FITZPATRICK, UND L. NATALE. **YARP: Yet Another Robot Platform.** *International Journal of Advanced Robotics Systems, special issue on Software Development and Integration in Robotics*, **3**(1), 2006. 36
- [95] QNX. **RTOS**. 37
- [96] C. BREAZEAL. **Kismet, the Robot**. 37
- [97] G. METTA, G. SANDINI, D. VERNON, L. NATALE, UND F. NORI. **The iCub Humanoid Robot: An Open Platform for Research in Embodied Cognition.** In *Proceedings of the 8th Workshop on Performance Metrics for Intelligent Systems, PerMIS '08*, Seiten 50–56, New York, NY, USA, 2008. ACM. 37
- [98] MICROSOFT. **Robotics Developer Studio**, 2012. 37, 77
- [99] KUKA LABORATORIES GMBH. **KUKA - Leichtbauroboter**. 37

- [100] WILLOW GARAGE. **Robot Operating System**. 37
- [101] WILLOW GARAGE. **PR2 - Personal Robot 2**. 37
- [102] SOUTHWEST RESEARCH INSTITUTE. **ROS for Industrial Applications**, 2012. 38
- [103] A. ELKADY UND T. SOBH. **Robotics Middleware: A Comprehensive Attribute-Based Bibliography**. *Journal of Robotics*, **2012**:15, 2012. 38
- [104] N. MOHAMED, J. AL-JAROUDI, UND I. JAWHAR. **Middleware for Robotics: A Survey**. In *Robotics, Automation and Mechatronics, 2008 IEEE Conference on*, Seiten 736–742, 2008. 38
- [105] OBJECT MANAGEMENT GROUP. **Common Object Request Broker Architecture (CORBA)**. 38
- [106] THE APACHE SOFTWARE FOUNDATION. **Etch**, 2013. 38
- [107] B. DUMAS, D. LALANNE, UND S. OVIATT. **Multimodal Interfaces: A Survey of Principles, Models and Frameworks**. In *Human Machine Interaction*, D. LALANNE UND J. KOHLAS, Editoren, **5440** von *Lecture Notes in Computer Science*, Seiten 3–26. Springer Berlin Heidelberg, 2009. 38, 40
- [108] R. SHARMA, V. PAVLOVIC, UND T. HUANG. **Toward multimodal human-computer interface**. *Proceedings of the IEEE*, **86**(5):853–869, 1998. 39
- [109] R. A. BOLT. **„Put-that-there“: Voice and gesture at the graphics interface**. In *Proceedings of the 7th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '80, Seiten 262–270, New York, NY, USA, 1980. ACM. 39
- [110] S. OVIATT. **Multimodal interactive maps: designing for human performance**. *Human-Computer Interaction*, **12**(1):93–129, März 1997. 40, 41
- [111] D. E. KNUTH. **Backus Normal Form vs. Backus Naur Form**. *Communications of the ACM*, **7**(12):735–736, Dezember 1964. 42, 83
- [112] NUANCE COMMUNICATIONS. **Dragon Naturally Speaking**. 42
- [113] CLT SPRACHTECHNOLOGIE. **DialogOS**. 42, 43
- [114] MICROSOFT. **Speech Recognition API Overview**. 42
- [115] G. SHIRES UND H. WENNBORG. **Web Speech API Specification**. 42
- [116] S. BIRD, E. KLEIN, UND E. LOPER. **Natural Language Processing with Python**. O'Reilly Media, June 2009. 42
- [117] WIT. **Natural language interface for mobile apps, home automation, wearable devices, robots and sms remote control**. 42
- [118] IVONA SOFTWARE SP. Z O.O. - AN AMAZON COMPANY. **IVONA - Text to Speech**, 2012. 43, 105

## LITERATURVERZEICHNIS

---

- [119] S. SOSNOWSKI, K. KUHNLENZ, UND M. BUSS. **EDDIE - An Emotion-Display with Dynamic Intuitive Expressions**. In *Robot and Human Interactive Communication, 2006. ROMAN 2006. The 15th IEEE International Symposium on*, Seiten 569–574, 2006. 43, 113, 114, 139, 141
- [120] P. EKMAN UND W. V. FRIESEN. **Constants across cultures in the face and emotion**. *Journal of personality and social psychology*, **17**(2):124, 1971. 43, 58
- [121] S. G. SOSNOWSKI. **Advances in Social and Underwater Robotics through Biomimetic Design**. Dissertation, Technische Universität München, München, 2013. 43
- [122] K. KÜHNLENZ, S. SOSNOWSKI, UND M. BUSS. **The Impact of Animal-like Features on Emotion Expression of Robot Head EDDIE**. *Advanced Robotics*, **24**(8-9), 2010. 43, 139
- [123] T. M. POITSCHKE. **Blickbasierte Mensch-Maschine Interaktion im Automobil**. Dissertation, Technische Universität München, München, 2011. 44
- [124] M. H. URBINA UND A. HUCKAUF. **Dwell-time free eye typing approaches**. In *Proceedings of the 3rd Conference on Communication by Gaze Interaction (COGAIN 2007)*, Seiten 65–70, 2007. 44
- [125] E. SCHNEIDER, T. VILLGRATTNER, J. VOCKEROTH, K. BARTL, S. KOHLBECHER, S. BARDINS, H. ULBRICH, UND T. BRANDT. **EyeSeeCam: An Eye Movement-Driven Head Camera for the Examination of Natural Visual Exploration**. *Annals of the New York Academy of Sciences*, **1164**(1):461 – 467, 2009. 45, 115, 139
- [126] D. A. ROBINSON. **A Method of Measuring Eye Movement Using a Scleral Search Coil in a Magnetic Field**. *Bio-medical Electronics, IEEE Transactions on*, **10**(4):137 –145, oct. 1963. 45
- [127] TOBII - EYETRACKING RESEARCH. **Eye Tracking Products Overview**. 45
- [128] F. BROZ, C. L. NEHANIV, D. K., UND H. KOSE BAGCI. **Towards Automated Human-Robot Mutual Gaze**. In *Proceedings The Fourth International Conference on Advances in Computer-Human Interactions, ACHI 2011*, Gosier, Guadeloupe, France, February 2011. 46, 59
- [129] SEEING MACHINES. **faceAPI**, 2010. 46, 60
- [130] M. SORIANO, B. MARTINKAUPPI, S. HUOVINEN, UND M. LAAKSONEN. **Skin detection in video under changing illumination conditions**. In *Pattern Recognition, 2000. Proceedings. 15th International Conference on*, **1**, Seiten 839–842 vol.1, 2000. 47
- [131] K. JOKINEN. **Gaze and gesture activity in communication**. In *Universal Access in Human-Computer Interaction. Intelligent and Ubiquitous Interaction Environments*, Seiten 537–546. Springer, 2009. 47
- [132] C. MAYER. **Facial Expression Recognition With A Three-Dimensional Face Model**. Dissertation, Technische Universität München, München, 2012. 47, 74
- [133] C. M. BISHOP. **Pattern Recognition and Machine Learning (Information Science and Statistics)**. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006. 48, 76

- [134] J. GAST, A. BANNAT, T. REHRL, C. MAYER, F. WALLHOFF, G. RIGOLL, UND B. RADIG. **Did I Get it Right: Head Gesture Analysis for Human-Machine Interaction.** In *Human-Computer Interaction. Novel Interaction Methods and Techniques*, Lecture Notes in Computer Science. Springer, 2009. 48, 143
- [135] T. REHRL. **Multimodale Mensch-Roboter-Interaktion für Ambient Assisted Living.** Dissertation, Technische Universität München, München, 2013. 48
- [136] RY MIX UND SIMBOUBOU. **Kinect SDK Dynamic Time Warping (DTW) Gesture Recognition**, 2011. 48
- [137] J. GAST, A. BANNAT, T. REHRL, F. WALLHOFF, G. RIGOLL, C. WENDT, S. SCHMIDT, M. POPP, UND B. FAERBER. **Real-time framework for multimodal human-robot interaction.** In *Human System Interactions, 2009. HSI '09. 2nd Conference on*, Seiten 276–283, may 2009. 49
- [138] D. MARLOW, S. KNICKERBOCKER, UND T. PLUNKETT. **Network Time Mechanisms for Improving Computer Clock Accuracy.** 49
- [139] M. GIULIANI, M. KASSECKER, S. SCHWÄRZLER, J. GAST, A. BANNAT, F. WALLHOFF, C. MAYER, M. WIMMER, C. WENDT, UND S. SCHMIDT. **MuDiS – A Multimodal Dialogue System for Human-Robot Interaction.** In *Proceedings of the 1st International Workshop on Cognition for Technical Systems*, 2008. 49
- [140] E. F. HILL. **Jess in Action: Java Rule-Based Systems.** Manning Publications Co., Greenwich, CT, USA, 2003. 50
- [141] A. COLMERAUER UND P. ROUSSEL. **The Birth of Prolog.** In *History of Programming languages—II*, T. J. BERGIN, JR. UND R. G. GIBSON, JR., Editoren, Seiten 331–367. ACM, New York, NY, USA, 1996. 50
- [142] SWI-PROLOG. **Robust, mature, free. Prolog for the real world.** 50
- [143] A. DENGEL. **Künstliche Intelligenz.** Bibliographisches Institut - Taschenbuchverlag, Mannheim, Leipzig, Wien, Zürich, 1994. 53
- [144] M. HAAS UND P. KRAMER. **Methoden künstlicher Intelligenz in betriebswirtschaftlichen Anwendungen.** Wismarer Schriften zu Management und Recht. Salzwasser-Verlag, 2009. 53
- [145] D. JAIN, L. MOSENLECHNER, UND M. BEETZ. **Equipping robot control programs with first-order probabilistic reasoning capabilities.** In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, Seiten 3626–3631. IEEE, 2009. 53
- [146] C. JUSTINE. **Nudge nudge wink wink: elements of face-to-face conversation for embodied conversational agents.** In *Embodied conversational agents*, Seiten 1–27. MIT Press, Cambridge, MA, USA, 2000. 55
- [147] C. L. KLEINKE. **Gaze and eye contact: a research review.** *Psychological bulletin*, **100**(1):78, 1986. 55

## LITERATURVERZEICHNIS

---

- [148] A. D. ENGELL UND J. V. HAXBY. **Facial expression and gaze-direction in human superior temporal sulcus.** *Neuropsychologia*, **45**(14):3234–3241, 2007. 55
- [149] S. LALLEE, K. HAMANN, J. STEINWENDER, F. WARNEKEN, U. MARTIENZ, H. BARRON-GONZALES, U. PATTACINI, I. GORI, M. PETIT, G. METTA, P. VERSCHURE, UND P. FORD DOMINEY. **Cooperative human robot interaction systems: IV. Communication of shared plans with Naïve humans using gaze and speech.** In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, Seiten 129–136, Nov 2013. 55
- [150] R. MROCYNSKI. **Grammatikalisierung und Pragmatikalisierung: zur Herausbildung der Diskursmarker wobei, weil und ja im gesprochenen Deutsch.** Tübinger Beiträge zur Linguistik. Narr, 2012. 56
- [151] B. MUTLU, T. SHIWA, T. KANDA, H. ISHIGURO, UND N. HAGITA. **Footing in Human-robot Conversations: How Robots Might Shape Participant Roles Using Gaze Cues.** In *Proceedings of the 4th ACM/IEEE International Conference on Human Robot Interaction, HRI '09*, Seiten 61–68, New York, NY, USA, 2009. ACM. 56
- [152] P. KULMS, N. C. KRÄMER, J. GRATCH, UND S.-H. KANG. **It's in Their Eyes: A Study on Female and Male Virtual Humans' Gaze.** In *Proceedings of the 10th International Conference on Intelligent Virtual Agents, IVA'11*, Seiten 80–92, Berlin, Heidelberg, 2011. Springer-Verlag. 56
- [153] L. ITTI UND C. KOCH. **Computational modelling of visual attention.** *Nature reviews neuroscience*, **2**(3):194–203, 2001. 56
- [154] M. ARGYLE UND M. COOK. **Gaze and mutual gaze.** Cambridge University Press Cambridge, Eng. ; New York, 1976. 56
- [155] S. ROTHGANGEL. **Kurzlehrbuch Medizinische Psychologie und Soziologie.** Georg Thieme Verlag, 2010. 57
- [156] S. MARTINEZ-CONDE, S. L. MACKNIK, UND D. H. HUBEL. **The role of fixational eye movements in visual perception.** *Nature Reviews Neuroscience*, **5**(3):229–240, 2004. 57
- [157] J. BATISTA, P. PEIXOTO, UND H. ARAUJO. **Real-time vergence and binocular gaze control.** In *Intelligent Robots and Systems, 1997. IROS '97., Proceedings of the 1997 IEEE/RSJ International Conference on*, **3**, Seiten 1348–1354 vol.3, 1997. 57
- [158] C. BREAZEAL, A. EDSINGER, P. FITZPATRICK, UND B. SCASSELLATI. **Active vision for sociable robots.** *Systems, Man, and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, **31**(5):443–453, September 2001. 57, 58
- [159] C. BREAZEAL. **Toward sociable robots.** *Robotics and Autonomous Systems*, **42**(3–4):167 – 175, 2003. Socially Interactive Robots. 57
- [160] J. RUESCH, M. LOPES, A. BERNARDINO, J. HORNSTEIN, J. SANTOS-VICTOR, UND R. PFEIFFER. **Multimodal saliency-based bottom-up attention a framework for the humanoid robot**

- iCub**. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, Seiten 962–967, 2008. 58
- [161] R. STIEFELHAGEN, H. K. EKENEL, C. FÜGEN, P. GIESELMANN, H. HOLZAPFEL, F. KRAFT, K. NICKEL, M. VOIT, UND A. WAIBEL. **Enabling Multimodal Human-Robot Interaction for the Karlsruhe Humanoid Robot**. *IEEE Transactions on Robotics*, **23**(5):840–851, 2007. 58, 59
- [162] C. L. SIDNER, C. D. KIDD, C. LEE, UND N. LESH. **Where to look: a study of human-robot engagement**. In *Proceedings of the 9th international conference on Intelligent user interfaces, IUI '04*, Seiten 78–84, New York, NY, USA, 2004. ACM. 58
- [163] S. P. LEE, J. B. BADLER, UND N. I. BADLER. **Eyes alive**. *ACM Transaction on Graphics*, **21**(3):637–644, Juli 2002. 59
- [164] C. PELACHAUD UND M. BILVI. **Modelling gaze behavior for conversational agents**. In *Intelligent Virtual Agents (IVA)*, 2003. 59
- [165] B. LANCE UND S. MARSELLA. **The Expressive Gaze Model: Using Gaze to Express Emotion**. *Computer Graphics and Applications, IEEE*, **30**(4):62–73, 2010. 59
- [166] J. STAHL. **Amplitude of human head movements associated with horizontal saccades**. *Experimental Brain Research*, **126**(1):41–54, May 1999. 62
- [167] J. AHLBERG. **CANDIDE-3 – An Updated Parameterised Face**. Technischer Bericht, Linköping University, 2001. 62, 63, 140
- [168] I. PANDZIC UND R. FORCHHEIMER. **MPEG-4 Facial Animation: The Standard, Implementation and Applications**. Wiley, 2003. 62
- [169] H.-C. NOTHDURFT. **Saliency and target selection in visual search**. *Visual Cognition*, **14**(4-8):514–542, 2006. 63
- [170] S. ONAT, K. LIBERTUS, UND P. KÖNIG. **Integrating audiovisual information for the control of overt attention**. *Journal of Vision*, **7**(10), 2007. 63
- [171] N. BUTKO, L. ZHANG, G. COTTRELL, UND J. MOVELLAN. **Visual saliency model for robot cameras**. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, Seiten 2398–2403, may 2008. 65
- [172] M. BINDEMANN, C. SCHEEPERS, UND A. BURTON. **Viewpoint and center of gravity affect eye movements to human faces**. *Journal of Vision*, **9**(2):1–16, 2009. 66
- [173] E. VATIKIOTIS-BATESON, I. EIGSTI, S. YANO, UND K. MUNHALL. **Eye movement of perceivers during audiovisual speech perception**. *Percept Psychophys*, **60**(6):926–40, 1998. 66
- [174] J. BLUME, T. REHRL, UND G. RIGOLL. **Multimodale Interaktion auf einer sozialen Roboterplattform**. *at – Automatisierungstechnik*, **61**(11):737–748, November 2013. 68, 118, 140

## LITERATURVERZEICHNIS

---

- [175] A. DIEKMANN. **Empirische Sozialforschung: Grundlagen, Methoden, Anwendungen**. Rowohlt's Enzyklopädie. Rowohlt-Taschenbuch-Verlag, 2007. 69
- [176] B. SCHULLER, R. MÜLLER, F. EYBEN, J. GAST, B. HÖRNLER, M. WÖLLMER, G. RIGOLL, A. HÖTHKER, UND H. KONOSU. **Being bored? Recognising natural interest by extensive audiovisual integration for real-life application**. *Image and Vision Computing*, **27**(12):1760–1774, November 2009. 74
- [177] S. KOHLBECHER, E. WIESE, K. BARTL, J. BLUME, A. BANNAT, UND E. SCHNEIDER. **Studying Gaze-based Human Robot Interaction: An Experimental Platform**. In *7th ACM/IEEE International Conference on Human-Robot Interaction, Boston, Massachusetts, USA*, 2012. 74
- [178] G. R. LEFRANCOIS. **Psychologie des Lernens (Springer-Lehrbuch)**. Springer, Berlin, 4. überarbeitete und erweiterte Edition, 2006. 75
- [179] LERNPSYCHOLOGIE. **Kognitivismus**. 75
- [180] U. LÄMMELE UND J. CLEVE. **Künstliche Intelligenz**. Carl Hanser Verlag GmbH & Company KG, 2012. 76
- [181] F. S. FOGLIATTO, G. J. DA SILVEIRA, UND D. BORENSTEIN. **The mass customization decade: An updated review of the literature**. *International Journal of Production Economics*, **138**(1):14–25, 2012. 77
- [182] C. EPPNER, J. STURM, M. BENNEWITZ, C. STACHNISS, UND W. BURGARD. **Imitation Learning with Generalized Task Descriptions**. In *Proceedings of the IEEE International Conference on Robotics and Automation*, Imitation Learning with Generalized Task Descriptions, Seiten 3968–3974, 12-17 May 2009. 77, 81, 82
- [183] J. NAKANISHI, J. MORIMOTO, G. ENDO, G. CHENG, S. SCHAAL, UND M. KAWATO. **Learning from demonstration and adaptation of biped locomotion**. *Robotics and Autonomous Systems*, **47**(2):79–91, 2004. 77
- [184] S. HESSE UND V. MALISA. **Robotik - Montage - Handhabung**. Carl Hanser Verlag, 2010. 77
- [185] KUKA ROBOTER GMBH. **KUKA Sim**. 77, 78, 140
- [186] C. MEYER, R. HOLLMANN, C. PARLITZ, UND M. HÄGELE. **Programmieren durch Vormachen für Assistenzsysteme „Schweiß- und Klebebahnen intuitiv programmieren“ (Programming by Demonstration for Assistive Systems „Intuitive Programming of Welding and Gluing Trajectories“)**. *it - Information Technology*, **49**(4):238–246, 2007. 78
- [187] PROF. DR.-ING. RÜDIGER DILLMANN. **Programmieren durch Vormachen in der Robotik**, 1994. 78
- [188] MRK-SYSTEME GMBH. **Safe Teaching**. 79, 140
- [189] M. HAUN. **Handbuch Robotik**. Springer Verlag, 2007. 79

- [190] M. PARDOWITZ, S. KNOOP, R. DILLMANN, UND R. ZOLLNER. **Incremental Learning of Tasks From User Demonstrations, Past Experiences and Vocal Comments.** *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, **37**(2):322–332, 2007. 80
- [191] M. N. NICOLESCU UND M. J. MATARIC. **Natural Methods for Robot Task Learning: Instructive Demonstrations, Generalization and Practice.** In *Proceedings of the Second International Joint Conference on Autonomous Agents and Multi-Agent Systems*, Seiten 241–248, 2003. 81
- [192] C. BREAZEAL, G. HOFFMAN, UND A. LOCKERD. **Teaching and working with robots as a collaboration.** In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems-Volume 3*, Seiten 1030–1037. IEEE Computer Society, 2004. 82
- [193] F. WALLHOFF, J. BLUME, A. BANNAT, W. RÖSEL, C. LENZ, UND A. KNOLL. **A skill-based approach towards hybrid assembly.** *Advanced Engineering Informatics*, **24**(3):329 – 339, August 2010. The Cognitive Factory. 83
- [194] A. BANNAT, J. GAST, G. RIGOLL, UND F. WALLHOFF. **Event analysis and interpretation of human activity for augmented reality-based assistant systems.** In *Intelligent Computer Communication and Processing, 2008. ICCP 2008. 4th International Conference on*, Seiten 1–8. IEEE, 2008. 84
- [195] JOINT ACTION OF HUMANS AND INDUSTRIAL ROBOTS. **Learning new Task using Instructions.** 86
- [196] JOINT ACTION OF HUMANS AND INDUSTRIAL ROBOTS. **Execution of previously learned Task.** 86
- [197] M. WAIBEL, M. BEETZ, J. CIVERA, R. D’ANDREA, J. ELFRING, D. GALVEZ-LOPEZ, K. HAUSSERMANN, R. JANSSEN, J. MONTIEL, A. PERZYLO, B. SCHIESSELE, M. TENORTH, O. ZWEIFLE, UND R. VAN DE MOLENGRAFT. **RoboEarth.** *Robotics Automation Magazine, IEEE*, **18**(2):69–82, 2011. 86, 130
- [198] GESELLSCHAFT DEUTSCHER VERSICHERER (GDV): TRANSPORT-INFORMATIONSSERVICE. **Fachbegriffe aus dem Bereich Verpackung**, 2012. 87
- [199] OPENCV. **Camera Calibration.** 89
- [200] H. KATO. **Augmented Reality Tool Kit.** 89, 140
- [201] D. MEAGHER. **Geometric modeling using octree encoding.** *Computer graphics and image processing*, **19**(2):129–147, 1982. 91, 140
- [202] R. B. RUSU, N. BLODOW, UND M. BEETZ. **Fast Point Feature Histograms (FPFH) for 3D Registration.** In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Kobe, Japan, May 12-17 2009. 96, 101
- [203] P. J. HUBER. **Robust estimation of a location parameter.** *Annals of Mathematical Statistics*, **35**(1):73–101, 1964. 97

## LITERATURVERZEICHNIS

---

- [204] Z. ZHANG. **Iterative point matching for registration of free-form curves and surfaces.** *International Journal of Computer Vision*, **13**(2):119–152, Oktober 1994. 101
- [205] PROFACTOR GMBH. **ReconstructMe**, 2012. 101
- [206] P. CHALMOVIANSKÝ UND B. JÜTTLER. **Filling Holes in Point Clouds.** In *Mathematics of Surfaces*, M. J. WILSON UND R. R. MARTIN, Editoren, **2768** von *Lecture Notes in Computer Science*, Seiten 196–212. Springer Berlin Heidelberg, 2003. 102
- [207] GOOGLE (MAINTAINED BY NEIL FRASER). **Blockly - A visual programming editor**, 2012. 104, 105, 141
- [208] P. PLAMONI. **Siri Proxy for Apple's Siri**, 2011. 105
- [209] METRALABS GMBH. **Scitos G5 – A professional mobile platform, combining the advantages of an industrial robot with the flexibility of a research robot.** 112
- [210] E. SCHNEIDER, S. KOHLBECHER, K. BARTL, F. WALLHOFF, UND T. BRANDT. **Experimental platform for Wizard-of-Oz evaluations of biomimetic active vision in robots.** In *Robotics and Biomimetics (ROBIO), 2009 IEEE International Conference on*, Seiten 1484–1489, 2009. 112
- [211] T. VILLGRATTNER UND H. ULBRICH. **Piezo-driven two-degree-of-freedom camera orientation system.** In *Industrial Technology, 2008. ICIT 2008. IEEE International Conference on*, Seiten 1–6, april 2008. 112
- [212] R. J. LEIGH UND D. S. ZEE. **The neurology of eye movements.** Nummer 55 in Contemporary neurology series. Oxford University Press, US, 3 Edition, 1999. 112
- [213] M. F. ZAEH, M. BEETZ, K. SHEA, G. REINHART, K. BENDER, C. LAU, M. OSTGATHE, W. VOGL, M. WIESBECK, M. ENGELHARD, C. ERTELT, T. RÜHR, M. FRIEDRICH, UND S. HERLE. **The Cognitive Factory.** In *Changeable and Reconfigurable Manufacturing Systems*, E. H. A., Editor, Springer Series in Advanced Manufacturing, Seiten 355–371. Springer London, 2009. 115
- [214] UNIVERSAL ROBOTS. **Collaborative Robot Solutions.** 116
- [215] MRK-SYSTEME GMBH. **KR 5 SI (SafeInteraction).** 116
- [216] FERROBOTICS COMPLIANT ROBOT TECHNOLOGY GMBH. **Roboter mit Gefühl**, 2012. 116
- [217] KEBA AG. **Automation by Innovation**, 2012. 116
- [218] V. VYATKIN. **IEC 61499 function blocks for embedded and distributed control systems design.** ISA-Instrumentation, Systems, and Automation Society, 2007. 116
- [219] D. BRŠČIĆ, M. EGGERS, F. ROHRMÜLLER, O. KOURAKOS, S. SOSNOWSKI, D. ALTHOFF, M. LAWITZKY, A. MÖRTL, M. RAMBOW, V. KOROPOULI, J. M. HERNÁNDEZ, X. ZANG, W. WANG, D. WOLLHERR, K. KÜHNLENZ, C. MAYER, T. KRUSE, A. KIRSCH, J. BLUME, A. BANNAT, T. REHRL, F. WALLHOFF, T. LORENZ, P. BASILI, C. LENZ, T. RÖDER, G. PANIN, W. MAIER, S. HIRCHE, M. BUSS, M. BEETZ, B. RADIG, A. SCHUBÖ, S. GLASAUER,

- A. KNOLL, UND E. STEINBACH. **Multi Joint Action in CoTeSys - setup and challenges**. Technischer Bericht, Technische Universität München & Ludwig-Maximilians-Universität München, Munich, Germany, 2010. 117, 141
- [220] C. LENZ, T. RÖDER, M. EGGERS, S. AMIN, T. KISLER, B. RADIG, G. PANIN, UND A. KNOLL. **A Distributed Many-Camera System for Multi-Person Tracking**. In *Proceedings of the First International Joint Conference on Ambient Intelligence (Aml 2010)*, R. WICHERT UND B. DE RUYTER, Editoren. Springer Lecture Notes in Computer Science, November 2010. 116
- [221] P. VIOLA UND M. JONES. **Rapid object detection using a boosted cascade of simple features**. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, Seiten 511–518, 2001. 117
- [222] ELOKENCE.COM SARL (ARNAUD MEGRET). **Akinator – das Web-Genie**. 118
- [223] B. GONSIOR, S. SOSNOWSKI, C. MAYER, J. BLUME, B. RADIG, D. WOLLHERR, UND K. KUHNLENZ. **Improving aspects of empathy and subjective performance for HRI through mirroring facial expressions**. In *Proceedings of the IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, Seiten 350–356. IEEE, 2011. 120
- [224] F. WALLHOFF, M. RUSS, G. RIGOLL, J. GÖBEL, UND H. DIEHL. **Improved Image Segmentation using Photonic Mixer Devices**. In *Proceedings of the International Conference on Image Processing (ICIP)*, Seiten 53–56. IEEE, 2007. 122
- [225] C. LENZ, M. RICKERT, G. PANIN, UND A. KNOLL. **Constraint Task-based Control in Industrial Settings**. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Seiten 3058–3063, St. Louis, MO, USA, 2009. 122
- [226] S. IHSEN, W. SCHNEIDER, F. WALLHOFF, UND J. BLUME. **Raising Interest of Pupils in Engineering Education Through Problem Based Learning**. *International Journal of Engineering Education*, **27**(4):789–794, 2011. 130
- [227] Q. PU, S. GUPTA, S. GOLLAKOTA, UND S. PATEL. **Whole-home Gesture Recognition Using Wireless Signals**. In *Proceedings of the 19th Annual International Conference on Mobile Computing & Networking, MobiCom '13*, Seiten 27–38, New York, NY, USA, 2013. ACM. 130
- [228] AUTOMATIONSPRAXIS. **Beginn einer neuen Ära**. 130
- [229] L. R. HOCHBERG, D. BACHER, B. JAROSIEWICZ, N. Y. MASSE, J. D. SIMERAL, J. VOGEL, S. HADDADIN, J. LIU, S. S. CASH, P. VAN DER SMAGT, ET AL. **Reach and grasp by people with tetraplegia using a neurally controlled robotic arm**. *Nature*, **485**(7398):372–375, 2012. 130
- [230] IIC. **Industrial Internet Consortium**. 130